Experimentation and Characterization of Mobile Broadband Networks

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electronics and Communications Engineering (30^{th}cycle)

Experimentation and Characterization of Mobile Broadband Networks

By

Ali Safari Khatouni

******

Supervisor(s):
Prof. Marco Mellia
Prof. Marco Ajmone Marsan

Doctoral Examination Committee:
Prof. Renato Lo Cigno, Referee, University of Trento
Prof. Marco Fiore, Referee, CNR IEIIT
Prof. Silvia Giordano, University of Applied Science - SUPSI
Prof. Marcelo Bagnulo Braun, Universidad Carlos III de Madrid
Prof. Matteo Cesana, Politecnico di Milano

Politecnico di Torino
2018

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Ali Safari Khatouni
2018

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I would like to dedicate this thesis to my loving Parisa and my parents

Acknowledgements

The opportunities that have been available to me as a graduate student at Politecnico di Torino have been unique, but without a doubt, the greatest opportunity of all has been the chance to work with my advisor Prof. Marco Mellia. What inspires me most about Marco aside from his well-known traits like superior intelligence, humility, and professional attitude is his never-ending passion for supporting his students.

I would also like to thank Marco for creating the perfect conditions for me to pursue my research ideas. I have learned many technical subjects from Marco. Besides, he has always inspired me by his endless patience and kindness to support me to move towards success in my professional career. The most important lesson for me was that you can be successful, supportive, kind, and "A SUPERB PROFESSOR". I will be grateful to him as long as I continue on my way. My sincere thanks also goes to my co-advisor Prof. Marco Ajmone Marsan for his insightful comments and encouragement. Without his precious support it would not be possible to conduct this research.

The atmosphere in laboratory plays an important role during these years; that is where I first met Luca, Danilo, Hassan, Enrico, Edilio, and Stefano. We spent really great time together and they helped me any time I needed their help. Then, Martino, Andrea (Mori), Leonardo, Michele, and Andrea joined us to build the complete 'Aula Esperti'. I am really happy to know them and spend these wonderful years with them. Beside my fellow labmates, I was really lucky and grateful to have part of my family, Hamid, Parima, Farmin, Golnaz, in Torino and Milano that gave me love and opportunity to spend quality time out of university.

My heart is full of gratitude for my parents and my brothers, who have always filled my life with unconditional love and support. I am utterly sorry and embarrassed for all the times I have been too “far” to return the love.

I finally want to thank my lovely wife and best friend, Parisa, who has the heart of an angel and who has supported me in all my life. She has helped me during all these years since the first time we met in Khayam Street in Urmia. I am also thankful to her for convincing me to keep a cute puppy, Romeo, that I spent all my lunch time with him.

Abstract

The Internet has brought substantial changes to our life as the main tool to access a large variety of services and applications. Internet distributed nature and technological improvements lead to new challenges for researchers, service providers, and network administrators. Internet traffic measurement and analysis is one of the most trivial and powerful tools to study such a complex environment from different aspects. Mobile BroadBand (MBB) networks have become one of the main means to access the Internet. MBB networks are evolving at a rapid pace with technology enhancements that promise drastic improvements in capacity, connectivity, and coverage, i.e., better performance in general.

Open experimentation with operational MBB networks in the wild is currently a fundamental requirement of the research community in its endeavor to address the need for innovative solutions for mobile communications. There is a strong need for objective data relating to stability and performance of MBB (e.g., 2G, 3G, 4G, and soon-to-come 5G) networks and for tools that rigorously and scientifically assess their performance. Thus, measuring end user performance in such an environment is a challenge that calls for large-scale measurements and profound analysis of the collected data. The intertwining of technologies, protocols, and setups makes it even more complicated to design scientifically sound and robust measurement campaigns. In such a complex scenario, the randomness of the wireless access channel coupled with the often unknown operator configurations makes this scenario even more challenging.

In this thesis, we introduce the MONROE measurement platform: an open access and flexible hardware-based platform for measurements on operational MBB networks. The MONROE platform enables accurate, realistic, and meaningful assessment of the performance and reliability of MBB networks. We detail the challenges we overcame while building and testing the MONROE testbed and argue

our design and implementation choices accordingly. Measurements are designed to stress performance of MBB networks at different network layers by proposing scalable experiments and methodologies. We study: (i) Network layer performance, characterizing and possibly estimating the download speed offered by commercial MBB networks; (ii) End users' Quality of Experience (QoE), specifically targeting the web performance of HTTP1.1/TLS and HTTP2 on various popular web sites; (iii) Implication of roaming in Europe, understanding the roaming ecosystem in Europe after the "Roam like Home" initiative; and (iv) A novel adaptive scheduler family with deadline is proposed for multihomed devices that only require a very coarse knowledge of the wireless bandwidth.

Our results comprise different contributions in the scope of each research topic. To put it in a nutshell, we pinpoint the impact of different network configurations that further complicate the picture and hopefully contribute to the debate about performance assessment in MBB networks. The MBB users web performance shows that HTTP1.1/TLS is very similar to HTTP2 in our large-scale measurements. Furthermore, we observe that roaming is well supported for the monitored operators and the operators using the same approach for routing roaming traffic. The proposed adaptive schedulers for content upload in multihomed devices are evaluated in both numerical simulations and real mobile nodes. Simulation results show that the adaptive solutions can effectively leverage the fundamental tradeoff between the upload cost and completion time, despite unpredictable variations in available bandwidth of wireless interfaces. Experiments in the real mobile nodes provided by the MONROE platform confirm the findings.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we present an introduction and outlook of this thesis. We briefly describe covered research topics. Then, we illustrate well-known scientific techniques used to solve these kinds of problems. Moreover, the structure of the thesis and a short guide are provided to ease the readability of the thesis.

1.1 Motivation and Research Questions

In this thesis, we aim at providing experimentation challenges and characterization of the Mobile BroadBand (MBB) [1] networks in mobile and stationary scenarios. Our goal is to take the user perspective using customized approaches as well as generic algorithms applied to wired and wireless network traffic processing. We focus on three major topics:

- Design and implementation of the first open access hardware-based platform for independent, multihomed, large-scale experimentation in MBB heterogeneous environments (MONROE)
- Performance assessment in MBB networks:
  - Speedtest-like measurement
  - Evaluation of users' Quality of Experience (QoE) [2] on the web domain
  - Understanding of roaming in Europe

- Profound study and evaluation of video upload scheduling problems with deadline

In the following, we provide a brief outlook of research questions which we intend to address.

1.1.1 Motivation

The Internet is one of the most sophisticated technologies that has ever been created by humans that has changed the way we live and communicate, allowing us to be informed, buy goods, enjoy shows, play games, keep in touch with friends, and freely express our opinions to potentially very large audiences. People are more and more connected to the Internet, with mobile terminals allowing access to information from anywhere and anytime. Considering this growth, there are many researchers trying to understand the Internet architecture, performance of different technologies, users' QoE, user behavior in the web, and etc.

Internet traffic measurement and monitoring are the most practical and powerful tools to study various aspects of the Internet and its effects on our live. The evolution of the Internet services and protocols has caused traditional traffic analysis approaches to be ineffective in certain cases. Traditional solutions for traffic analysis, classification, and measurement fall short in fully understanding of Internet services and protocols as a key requirement for network monitoring/planning and security monitoring tools. When it comes to MBB networks, the picture becomes much more complicated than wired networks because of several additional factors.

MBB networks have become a crucial infrastructure for people to stay connected everywhere and while on the move. Society's increased dependence on MBB networks motivates researchers and engineers to enhance the capabilities of mobile networks by designing new technologies to cater for a plenty of new applications and services, growth in traffic volume and a wide variety of user devices (e.g., smart phones, tablets, smart devices, toys, etc.). Wireless technologies such as WiFi, 2G, 3G, 4G, and soon-to-come 5G, provide access capacities up to hundreds of Mb/s. Still, there are scenarios in which the volume of data being produced and consumed challenges the bandwidth offered by wireless networks.

Subsequently, researchers show a regularly revived interest in understanding the dynamics of such an emerging environment. A rigorous knowledge of the services of MBB network providers would allow a better network administration, an enhanced resilience against failures, an intelligent usage of available resources, an appropriate support of new applications, and the provisioning of new services. Such notions are preeminent today as the MBB network is mostly compelled by economic interests: Technical achievements bring improvements over previous technologies but their actual adoption depends on the usefulness to develop new businesses. In addition, rising attention is given to the QoE offered to end users: given the competitive reality in which they work, Internet Service Providers (ISP), and MBB network providers must always provide a satisfactory QoE.

To measure the network accurately and fairly, it is crucial to identify the metrics that accurately capture the performance and the conditions under which these parameters should be measured. These parameters might be different for various stakeholders. For example, regulators need connectivity, coverage and speed information collected from a third-party, an independent platform to monitor how operators fulfill their obligations, and a baseline for designing regulatory policies. On the other hand, operators are interested in operational instability and anomalies to identify problems in their networks. For end-users, Quality of Service (QoS) and QoE parameters are of paramount importance, while application developers need information about the underlying network to design robust services and protocols. There are only limited studies in the literature that focus on identifying these parameters. Therefore, one of the main objectives and unique features of this thesis is its potentials to define measurement methodologies and to experimentally verify them in order to accurately reflect the performance and reliability of MBB networks from the perspective of different stakeholders.

Achieving an extensive understanding of MBB network dynamics is not a straightforward task. There is strong need to develop methodologies that are not case-specific or dependent on the application. To this end, we aim to design and build platforms and methodologies able to infer profound statistics by leveraging the network traffic and different visibility aspects of operational MBB networks. We propose to build an open measurement platform to inspect MBB network traffic at different levels, e.g., per-packet, per-flow, per-user, per-application, in order to gain a complete understanding of MBB networks.

In this complex environment, there is a substantial need for both open data about the performance and reliability of commercial operators, as well as open platforms for experimentation with operational MBB providers. Thorough methodological repeatable end-to-end measurements are vital for evaluating network performance, determining the quality experienced by end users and analyzing novel protocols. Existing experimental platforms, such as Planetlab [3], RIPE Atlas [4], or CAIDA Ark [5]; meet these requirements. However, they are limited to fixed broadband networks and are not multihomed.

To this end, we introduce MONROE measurement platform, uses to accurately identify key performance indexes and then allow experimenters to measure and experiment with them as realistically as possible.

1.2 Topics Outline

1.2.1 MONROE

In this thesis, we introduce MONROE: the first open access hardware-based platform for independent, multihomed, large-scale experimentation in MBB heterogeneous environments. MONROE platform enables accurate, realistic and meaningful assessment of the performance and reliability of 11 MBB networks in Europe. We report on our experience designing, implementing and testing our proposed solution for the platform. We detail the challenges we overcame while building and testing the MONROE testbed and discuss our design and implementation choices accordingly. Ultimately, we describe and exemplify the capabilities of the platform.

A typical alternative to using controlled testbeds such as MONROE is to rely on end users and their devices to run tests by visiting a website [6] or running a special application [7]. The main advantage of such crowdsourcing techniques is scalability: it can collect millions of measurements from different regions, networks, and user equipment types [8]. However, repeatability is challenging and one can only collect measurements at users' own will, with no possibility of either monitoring or controlling the measurement process. Mostly due to privacy reasons, crowd measurements do not always provide important context information (e.g., location, type of user equipment, type of subscription, and connection status, e.g., 2G, 3G, 4G, or WiFi). MONROE is complementary to crowdsourcing approaches and the control

over the measurement environment tackles the drawbacks of crowd data, though at the cost of a smaller geographical footprint [9]. Furthermore, MONROE supports the deployment of different applications and protocols, and enables benchmarking tools and methodologies.

1.2.2 MBB Networks Performance Assessment

MBB networks revolutionized the way people interact and brings a variety of communication services into most of our daily activities. Today, messaging, videos, and the web are key components of our lives, and we expect our MBB network providers to supply high performance service in extremely dynamic scenarios.

When coming to performance assessment, the picture becomes much more complicated in MBB networks than in wired networks. Even the simplest of the tests, i.e., a "speedtest-like" measurement of the single TCP [10] bulk download speed using HTTP, may become complicated to interpret in MBB networks, due to the large number of factors that affect performance. Physical impairments, mobility, variety of devices, presence of Performance Enhancing Proxies (PEP) [11], different access network configurations, etc., all possibly impact the measurement results, and complicate the picture.

Speedtest-like Measurements in MBB Networks

In the first part, we report our experience in designing, running, and analyzing speedtest-like experiments on MONROE nodes. Despite the large dataset, and the scientific approach, we find that running even a simple speedtest-like experiment proves to be very complicated, with results that apparently vary on a large scale, with no obvious correlations, and sometimes in an unpredictable way. We observe the presence of Network Address Translation (NAT), and of transparent proxies, as well as different access network configurations, and roaming agreements, each adding complexity to the already complicated picture.

`WebWorks`: Experimenting the Mobile Web

The intertwining of technologies, protocols, setups, and service design makes it complicated to design scientifically sound and robust measurement campaigns. For example, the higher the load in the MBB network cell, the larger the variance users perceive in the time to reach the content of interest (e.g., a webpage), which, in turn, translates into poor QoE. In this complex ecosystem, data analytics that focus on finding relationships between user experience and network performance statistics offer the promise of helping operators target those technology improvements that matter most to their customers.

In this part, we discuss different ways to monitor service performance in MBB networks, with the objective of quantifying end user QoE in web while using HTTP1.1/TLS and HTTP2 with focusing on various popular web sites. We exploit the MONROE system, which we built to enable controlled experiments in multiple MBB networks under similar conditions. Then, by leveraging data analytics, we show how the data we collected enables us to directly relate user experience to network performance statistics, an important step on the way to monitoring and managing service quality and user satisfaction.

Understanding Roaming in Europe

International roaming allows mobile users to use their voice and data services when they are abroad. The European Commission (EC), in an effort to create a single digital market across the European Union (EU), has recently (as of June 2017) introduced a set of regulatory decisions [12] as part of the "Roam like Home" initiative. This initiative abolishes additional charges for users when they use voice and data services while roaming in EU. In this setting, MNOs are expected to deliver services with QoS properties similar to the ones a user experiences when at home. As a result, people are able to use data services more freely across Europe. However, the performance implications of roaming solutions have not been carefully examined.

This work provides an in-depth characterization of the implications of international data roaming within Europe. We build a unique roaming measurements platform using 16 different mobile networks deployed in 6 countries across Europe. Using this platform, we measure different aspects of international roaming in MBB

networks, including mobile network configuration, performance characteristics, and content discrimination.

1.2.3 Deadline-Constrained Content Upload from Multihomed Devices

Our interest is motivated and inspired by the real needs of public transport operators. Public transport vehicles (like buses or trains) are equipped with multiple MBB interfaces; several onboard security cameras record videos. Those must be uploaded to a security center where an operator occasionally requests to watch selected portions of the videos. In this scenario, continuous real-time video uploading is too expensive. Even if current MBB networks can offer capacities up to 100 Mb/s, the number of vehicles and videos, the limited data quota, the performance variability along the route, and the need to check only parts of the videos, call for ingenious upload strategies. Hence, videos are stored onboard, and, only when an alarm is triggered, the security operator on duty requests the specific portion of the video that must be uploaded before a specified short deadline.

In this thesis, we propose and analyze a family of adaptive schedulers that require only a very coarse knowledge of the available bandwidth on wireless interfaces. The main contributions of this research topic are: i) Devising mathematical formulations of the deadline constrained content upload problem from multihomed terminals, under different assumptions; ii) Reporting extensive evaluations of the proposed solutions, based on trace-driven simulations using recently collected traces; and iii) Designing, implementing, testing, and evaluating a real implementation of the proposed dynamic algorithm on MONROE nodes.

1.3 Internet Measurement Techniques and tools

The research work realized in this thesis has its roots in Internet measurement platform [13] and scientific approach to use these platforms for Internet traffic measurement [14]. Internet is one of the most sophisticated technologies that has ever been created by humans and it is evolving rapidly. Internet traffic measurement and analysis is one of the most powerful tools to understand this phenomenon.

These days, Internet measurement platform plays a vital role in running large-scale measurements to gain insight into operators operational setting, Internet service, user QoE, and etc. When it comes to MBB networks, obtaining such information needs more efforts and well studied approach because of the nature of MBB networks.

In this section, we present description of the scientific techniques that we exploit in this thesis. These consist of either specialized techniques for network traffic processing or general purpose algorithms applied to specific contexts. In the latter case, additional details are reported along with the dissertation of the research topic for which they are applied.

1.3.1 Passive Traffic Measurement

Passive traffic collection is a tool aimed at storing network traffic as it passes through communication media and devices ideally without any change or interfering the traffic. Capturing traffic needs to setup a tool (i.e., probe) instrumented to sniff raw packets while they are passing through the network. Probes can capture traffic either at packet level or flow level; in the latter case there is need to process packets as they capture and extract information about layer-3 or layer-4 flows. Probes can be either specialized hardware or be built into already existing network devices.

Passive measurements are the most practical means to analyze the real behavior of network users. They provide immediate and detailed insights about the actual usage of the network at the physical layer. Furthermore, when proper processing is performed on captured data, it can provide higher level metrics to measure, e.g., users' perceived QoE [15], video streaming quality, and etc. The collected data includes knowledge about the users and services that they are using.

Fig. 1.1 shows a typical deployment of network passive monitoring at ISP level. A probe seats at ISP level, e.g., a Point of Presence (PoP) where households' traffic is aggregated. All users' connections behind the probe can be possibly captured and analyzed by the probe. Moreover, nowadays probes are able to filter connections or packets with protocol-based and content-based filtering rules.

In this thesis, passive measurements are employed to characterize the MBB networks performance assessment (Chapter 3), to assess users' QoE in case of using different applications (Chapter 4), to understand the roaming in Europe (Chapter 5), and to create traffic traces to present the behavior of operational MBB networks

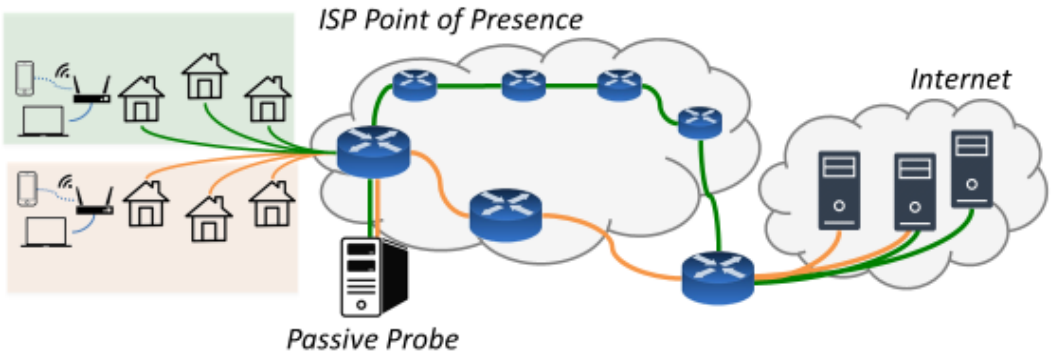


Fig. 1.1 Typical network probe deployment at ISP level

(Chapter 6). To this end, we use TCP STatistic and Analysis Tool (`Tstat`[1] [16]) to get flow level information, we also exploit `tcpdump`[2] to capture packet level traces.

Tstat

Tstat is an advanced open-source tool for passive network traffic monitoring and analysis. Tstat is a high-performing passive probe able to monitor live networks up to 40 Gb/s speed on off-the-shelf hardware [17]. It also brings traffic classification capabilities through behavioral classifiers [18, 19], high-level visibility on encrypted traffic through the analysis of Domain Name System (DNS) queries [20], and a thorough characterization of activities in the monitored network.

The basic operation performed by Tstat consists of processing the IP packets passing on the link to rebuild upper network layer flows. Packets group according to precise rules that define a flow identifier. A conventional choice is to aggregate packets according to a tuple defined by (L4 protocol, source IP address, source port, destination IP address, destination port). For TCP, the beginning and the end of a TCP flow are provided by the identification of the connection set-up and tear-down messages, i.e., SYN and FIN flags set in the TCP header, respectively. In case the connection is unexpectedly interrupted without the FIN messages, the flow is considered closed after an idle time. For UDP, a flow is identified when the first packet matching a new flow identifier and considered closed after an idle time.

[1] http://tstat.polito.it/
[2] http://www.tcpdump.org/

Tstat provides a valuable set of statistics, some of which are common to all flows, e.g., source and destination IP addresses, timestamp of the first and last packet seen, number of bytes and packets exchanged, and connection duration. Other statistics instead depend on the L4 protocol. While for UDP only the source and destination port numbers are reported, TCP statistics are more than 100 different metrics such as counters for TCP flags, i.e., SYN, ACK, FIN, RST, timestamps for first and last packet with payload, and number of retransmitted bytes and packets, etc.

1.3.2 Active Traffic Measurement

Active traffic measurement refers to a technique in which we inject traffic into the network and study the behavior of the network with respect to the injected traffic. Typically, they are used to provide network performance statistics, e.g., checking connectivity, packet loss, path changes, and etc. Usually, each active tool is designed to address a specific problem and there is possibility for researchers to design their own customized active tools.

However, these techniques bring extra cost. We need to run an experiment which means inserting additional traffic into network. It causes more load on the networks that can affect the users in the network and create congestion in network. Moreover, active measurement can be trickier in some cases. For instance, we still pay charges based on the traffic volume usage in the MBB networks, thus it calls for efficient and careful use of active measurement in these environments.

In this thesis, active measurements are used to assess MBB networks performance (Chapter 3, 4, 5, and 6). To this end, we use several classical tools (e.g., `traceroute` [21], `ping` [22] and `iperf` [23]) and customized tools that we will describe in following chapters.

1.3.3 Optimization Technique

The concept of optimization is rooted in a basis underlying the analysis of a wide range of allocation problems or complex decisions. It provides a degree of philosophical elegance that is hard to dispute. In complex decision problems, involving the selection of values for a number of interrelated variables, by concentrating on a single objective designed to quantify performance and measure the quality of the

decision. This objective is minimized (or maximized, depending on the problem or the formulation) subject to the constraints that may narrow the selection of decision variable values. If a proper single aspect of a problem can be isolated and characterized by an objective, optimization may provide an appropriate framework.

For instance, we present a very simple problem in which the goal is minimizing the production costs of a company. The company builds B different products and each item i has cost equal to c_i. The company can only build C items in total. We present a possible formulation as following:

$$\textbf{min} \sum_{i}^{B} c_i x_i \tag{1.1}$$

$$\sum_{i}^{B} x_i < C \tag{1.2}$$

$$x_i \geq 0 \;\; \forall i \in [0, B] \tag{1.3}$$

The objective function in (1.1) presents the total cost, which must be minimized. In our formulation, x_i is the decision variable that we want to figure out to minimize the total cost, i.e., the number of items needed to produce from product i. Expressions (1.2 and 1.3) force constraints in problem. Expression (1.2) states that the sum of decision variable is less than constant value C, i.e., total number of production of all products. Expression (1.3) forces the decision variable and can not be negative, i.e., the number of items can not be negative.

It is a scarce problem in which it is feasible to fully represent all the complexities of variable interactions, constraints, and appropriate objectives when dealing with a sophisticated problem. Therefore, an approximation of a specific optimization formulation should be considered. Ability in modeling, reasonable interpretation of results, and in getting the important aspects of a problem are needed to acquire purposeful conclusions. Problem formulation itself involves a trade-off between the conflicting objectives of building a mathematical model to precisely capturing the problem specification and building a model that is tractable.

In this thesis, optimization methods are used to create a model for video upload scheduling in multihomed systems (Chapter 6). To this end, we exploit different kinds of formulations, e.g., Linear programing, Stochastic optimization, and customized heuristics that we will describe in the following chapter.

1.3.4 Statistical Correlation Method

Correlation is a statistical method used to assess a possible linear association between two variables. It is simple both to calculate and to interpret. There are two major types of correlation coefficients: Pearson's product moment correlation coefficient [24] and Spearman's rank correlation coefficient [25]. The correct usage of correlation coefficient type depends on the types of variables being studied.

The most popular type of correlation coefficient is Pearson. Informally, it can be said that the correlation coefficient demonstrates the extent to which values of two variables are "proportional" to each other. The value of the correlation (i.e., correlation coefficient) does not depend on the particular measurement units used. Proportional means linearly related; that is, the correlation is high if it can be approximated by a straight line (sloped upwards or downwards). This line is called the regression line or least squares line, because it is determined in such a way that the sum of the squared distances of all the data points from the line is the lowest possible. Pearson correlation assumes that the two variables are measured on at least interval scales.

Pearson's correlation coefficient when applied to a sample is commonly represented by the letter r and may be referred to as the sample correlation coefficient or the sample Pearson correlation coefficient. We can obtain a formula for r by substituting estimates of the covariances and variances based on a sample into the formula above. Expression 1.4 illustrates the Pearson product moment correlation coefficient calculation, for one dataset $x_1, ..., x_n$ containing n values and another dataset $y_1, ..., y_n$ containing n values.

$$r = \frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=0}^{n}(y_i - \bar{y})^2}} \tag{1.4}$$

Where: n is the number of samples, x_i and y_i are the single samples indexed by i, and $\bar{x} = \frac{\sum_{i=0}^{n} x_i}{n}$.

In this thesis, correlation method is used to verify correlation between different metrics collected at network layer and physical layer in MBB networks (Chapter 3).

1.3.5 Statistical Distance Measure

There are statistical approaches to compactly quantify the difference between two distributions. Statistical distance quantifies the distance between two statistical objects, which can be two random variables or two probability distributions. In formal terms, the comparison function has the form $F(p,q) : (\mathbb{R},\mathbb{R}) \rightarrow \mathbb{R}$, while the quantization function can be defined as $Q(F(p,q)) : \mathbb{R} \rightarrow \mathbb{N}$, where $p = p(x)$ and $q = q(x)$ are two empirical distributions under analysis.

To simplify the discussion, we consider a single statistical distance measure (SDM) for the sake of illustration, and defer to the Appendix A through discussion and sensitivity analysis on all related settings (e.g., different metrics, population size, binning, etc.). As a representative SDM in this class, we take the Jensen-Shannon divergence (JS_{div}), which is defined as:

$$JS_{div} = \sum_i \Big\{ \frac{1}{2} p_i \ln\Big(\frac{p_i}{\frac{1}{2}p_i + \frac{1}{2}q_i}\Big) + \frac{1}{2} q_i \ln\Big(\frac{q_i}{\frac{1}{2}q_i + \frac{1}{2}p_i}\Big) \Big\}$$

where p_i and q_i are the empirical probabilities of samples taking values in the i-th bin. JS_{div} is a popular statistical measure based on the Kullback-Leibler divergence. JS_{div} adds symmetry, i.e., $JS_{div}(p,q) = JS_{div}(q,p)$, and bounded image, i.e., $JS_{div} \in [0, \ln(2)]$ to the Kullback-Leibler divergence. JS_{div} is equal to 0 if $p = q$, while it saturates to $\ln(2)$ for two completely disjoint distributions.

In our context, we specifically look for SDMs with bounded support as it makes the comparison of the difference between distributions more practical. More importantly, the symmetry property is required as it makes the SDM invariant to the choice of the distribution considered as reference. While asymmetric metrics can be used to contrast a suspect population against a well-behaving one, we have no apriori knowledge on which population should be considered the reference.

While we discuss these issues further in Appendix A, the information provided in this section allows us to understand the application of the general framework in users' QoE comparison, we focus on it in the Chapter 4.

1.4 Thesis Organization

Besides this introduction, the thesis is organized into six chapters.

Chapter 2 (MONROE) focuses on presentation of MONROE: the first open access hardware-based platform for independent, multihomed, large-scale experimentation in MBB heterogeneous environments. We describe MONROE in detail and its architecture and capabilities. Most of this work has its roots in the following papers:

- O. Alay, A. Lutu, R. Garcia, M. Peon Quiros, V. Mancuso, T. Hirsch, T. Dely, J. Werme, K. Evensen, A. Fosselie Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, **A. Safari Khatouni**, M. Mellia, M. Ajmone Marsan, R. Monno, H. Lonsethagen, Measuring and Assessing Mobile Broadband Networks with MONROE, 17th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 21-24, 2016

- O. Alay, A. Lutu, R. Garcia, M. Peon Quiros, V. Mancuso, T. Hirsch, T. Dely, J. Werme, K. Evensen, A. Fosselie Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, **A. Safari Khatouni**, M. Mellia, M. Ajmone Marsan, R. Monno, H. Lonsethagen, Demo: MONROE, a distributed platform to measure and assess mobile broadband networks, ACM WiNTECH, October 3, 2016

- O. Alay, A. Lutu, M. Peon-Quir, V. Mancuso, T. Hirsch, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, **A. Safari Khatouni**, M. Mellia, M. Ajmone Marsan, Experience: An Open Platform for Experimentation with Commercial Mobile Broadband Networks, MobiCom, the 23th Annual International Conference on Mobile Computing and Networking, October 16-20, 2017

Chapter 3 (Speedtest-like Measurements in MBB Networks) concentrates on "speedtest-like" measurements to evaluate the download speed offered by MBB networks. We indicate that the benchmarks for the performance assessment of MBB networks are needed, in order to avoid simplistic, superficial, wrong, or even biased studies, which are difficult to prove false.

Most of this work has its roots in the following paper:

- **A. Safari Khatouni**, M. Mellia, M. Ajmone Marsan, S. Alfredsson, J. Karlsson, A. Brunstrom, O. Alay, A. Lutu, C. Midoglu, V. Mancuso, Speedtest-like Measurements in 3G/4G Networks: the MONROE Experience, 29th International Teletraffic Congress, September 4-8, 2017

Chapter 4 (`WebWorks`: Experimenting the Mobile Web) focuses on users' QoE of web on commercial mobile carriers. Our results and further analysis shed light on the complexity of the cellular networks, where the randomness of the wireless access channel coupled with the often unknown operator configurations makes monitoring performance very challenging. We find that the overall web performance is similar across different countries and operators, with only slight variations. In aggregate per target websitse, our measurements show that the performance improvements HTTP2 promised still remain to be experienced.

Most of this work has its roots in the following paper:

- M. Peon-Quiros, V. Mancuso, V. Comite, A. Lutu, O. Alay, S. Alfredsson, J. Karlsson, A. Brunstrom, M. Mellia, **A. Safari Khatouni**, T. Hirsch, Results from running an experiment as a service platform for mobile networks, The 11th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, October 20, 2017

Chapter 5 (Understanding Roaming in Europe) focuses on an profound characterization of the implications of international data roaming within Europe. We opted for a unique roaming measurements platform using 16 different mobile networks deployed in 6 countries across Europe. Using this platform, we measure different aspects of international roaming in MBB networks, including mobile network configuration, performance characteristics, and content discrimination. Results show that operators adopt common approaches to implementing roaming.

Most of this work has its roots in the following paper:

- **Submitted:** A. M. Mandalari, A. Lutu, A. Custura, **A. Safari Khatouni**, O. Alay, M. Bagnulo, V. Bajpai, A. Brunstrom, J. Ott, M. Mellia, G. Fairhurst, Experience: Implications of Roaming in Europe, the 24th Annual International Conference on Mobile Computing and Networking

Chapter 6 (Deadline-Constrained Content Upload from Multihomed Devices) focuses on the work originating from the practical requirements of video surveillance in public transport systems, where security cameras store video onboard, and a central operator occasionally needs to access portions of the recordings. When this happens, the selected video portions must be uploaded within a given deadline,

using (multiple) wireless interfaces, with different costs (which correspond to, e.g., tariffs). We study this video upload problem as a scheduling problem with deadline, where our goal is to choose which interfaces to use and when, so as to minimize the cost of the upload while meeting the given deadline. Our study gives rise to adaptive schedulers that require only a very coarse knowledge of the wireless interfaces bandwidth.

Most of this work has its roots in the following papers:

- **A. Safari Khatouni**, M. Ajmone Marsan, M. Mellia, Video Upload from Public Transport Vehicles using Multihomed Systems, 2016 IEEE Conference on Computer Communications Workshops: Student Activities, 10 April 2016
- **A. Safari Khatouni**, M. Ajmone Marsan, M. Mellia, Delay Tolerant Video Upload from Public Vehicles, Smart Cities and Urban Computing, April 11, 2016
- **A. Safari Khatouni**, M. Ajmone Marsan, M. Mellia, R. Rejaie, Adaptive Schedulers for Deadline-Constrained Content Upload from Mobile Multihomed Vehicles, The 23rd IEEE International Symposium on Local and Metropolitan Area Networks, June 12-14, 2017
- **Submitted: A. Safari Khatouni**, M. Ajmone Marsan, M. Mellia, R. Rejaie, Deadline-Constrained Content Upload from Multihomed Devices: Formulations and Algorithms, Computer Networks (COMNET), 2018

Chapter 7 summarizes this work, recaps the collected findings, and highlights the most significant results obtained. In addition, **Appendix A** presents additional detail that are kept out of the main flow of this work to improve readability. Finally, **Appendix B** illustrates a short biography of the author.

1.5 Readers' Guide

I have actively participated in MONROE project during my PhD. Thus, some parts of this thesis was developed in collaboration with other partners and researchers. There are several parts of this thesis for which I have developed the main ideas and

methodologies on my own. In the following, I list the parts whose the main ideas and methodologies have been done by myself:

- Integration of the Tstat passive probe in the MONROE platform with design, implementation, and support of corresponding backend systems (Section 2.4)
- The idea, methodology, and analysis developed in Chapter 3
- Methodology and statistical analysis presented in Section 4.4.2
- Besides the experiment coordination, experiment design and analysis of collected data in Section 5.4.1
- The idea, methodology, and analysis developed in Chapter 6

The rest of the work has been done in collaboration with other researchers (see Appendix B).

Chapter 2

MONROE

2.1 Introduction

MBB networks have become the key infrastructure for people to stay connected everywhere they go and while on the move. Society's increased reliance on MBB networks motivates researchers and engineers to enhance the capabilities of mobile networks by designing new technologies to cater for a plethora of new applications and services, growth in traffic volume and a wide variety of user devices. In this dynamic ecosystem, there is a strong need for both open objective data about the performance and reliability of commercial operators, as well as open platforms for experimentation with operational MBB providers.

In this thesis, we introduce MONROE: the first open access hardware-based platform for independent, multihomed, large-scale experimentation in MBB heterogeneous environments. MONROE comprises a large set of custom hardware devices, both mobile (e.g., via hardware operating aboard public transport vehicles) and stationary (e.g., volunteers hosting the equipment in their homes), all multihomed to three operators using commercial grade subscriptions.

Thorough systematic repeatable end-to-end measurements are essential for evaluating network performance, assessing the quality experienced by end users and experimenting with novel protocols. While existing experimental platforms, such as PlanetLab [3], RIPE Atlas [4] or CAIDA Ark [5], meet these requirements, they are limited to fixed broadband networks and are not multihomed. MONROE is a one-of-a-kind platform that enables controlled experimentation with different com-

mercial mobile carriers. It enables users to run custom experiments and to schedule experimental campaigns to collect data from operational MBB and WiFi networks, together with full context information (metadata). For example, MONROE can accommodate performance evaluation of different applications (e.g., web and video) over different networks or testing various protocols and solutions under the same conditions.

Objective performance data is crucial for regulators to ensure transparency and the general quality level of the basic Internet access service [26]. Several regulators responded to this need with ongoing nationwide efforts [27]. Often, they do not open the solutions to the research community to enable custom experimentation, nor do they grant free access to the measurement results and methodology. MONROE aims to fill this gap and offers free access to custom experimentation. The MONROE project selected different external users to deploy their own custom experiments on the MONROE system with the purpose of testing and further improving the platform based on their feedback.

A common alternative to using controlled testbeds such as MONROE is to rely on end users and their devices to run tests by visiting a website [6] or running a special application [7]. The main advantage of such crowdsourcing techniques is scalability: it can collect millions of measurements from different regions, networks and user equipment types [8]. However, repeatability is challenging and one can only collect measurements at users' own will, with no possibility of either monitoring or controlling the measurement process. Mostly due to privacy reasons, crowd measurements do not always provide important context information (e.g., location, type of user equipment, type of subscription, and connection status (2G/3G/4G and WiFi)). MONROE is complementary to crowdsourcing approaches and the control over the measurement environment tackles the shortcomings of crowd data, though at the cost of a smaller geographical footprint [9]. Furthermore, MONROE supports the deployment of different applications and protocols, and enables benchmarking tools and methodologies.

In the rest of the chapter, we report on our experience designing, implementing and using the platform. We detail the design considerations and demonstrate the versatility of our approach (Section 2.2). We explain how we cater for the requirements of experimenters and enable them to deploy myriad measurements on operational commercial MBB networks. The MONROE measurement node (hereinafter, the

node or the MONROE node) sits in the center of the system and is the most important element, conditioning the proper functionality of the measurement system. We describe our experience with the MONROE system implementation and detail the hardware selection for the MONROE measurement node (Section 2.3). We forged the node to be flexible and powerful enough to run a wide range of measurement and experimental tasks, including demanding applications like adaptive video streaming. In the same time, we ensured that the node software design translates into a robust implementation (Section 2.4) that is also easily evolved and upgraded in order to sustain the most recent technological innovations. We further present the user access and scheduling solution we offer experimenters for exploiting the available resources of the platform in a fair manner (Section 2.5).

We show that the MONROE system is a fitting solution to conduct a wide range of experiments over commercial cellular networks. To showcase its capabilities, we describe different categories of experiments MONROE supports (Section 2.6), which give an overview of the main categories of experiments MONROE are conducting at the time of writing. Finally, Section 2.7 concludes the chapter.

2.2 System Design

Throughout the design process of MONROE, we interacted with the users of the platform (e.g., universities, research centers, industry and SMEs[1]) and collected their feedback on requirements for platform functionality. This allowed us to gauge experimenters' expectations and use them to sketch the platform specifications.

2.2.1 Requirements

We summarize the main requirements as follows.

Large scale and Diversity: To give a representative view of the characteristics of an entire network, we need to collect measurements from a large number of vantage points. Furthermore, we should strive to collect measurements under diverse geographical settings, from major cities to remote islands.

[1] Small and medium-sized enterprises

Mobility: Mobility is what makes MBB networks unique compared to other wireless networks. To provide insight into the mobility dimension of MBB networks, it is imperative that the platform integrates a deployment under realistic mobility scenarios.

Fully programmable nodes: To accommodate the wide range of experiments users contemplate to run on the platform, we should forge measurement devices that are flexible, powerful and robust.

Multihoming support: To compare different mobile operators and/or different wireless technologies under the same conditions, the same node should connect to multiple providers at the same time (multihoming support). This further makes the platform particularly well suited for experimentation with methods that exploit aggregation of multiple connections.

Rich context information: While analyzing the measurements, context information is crucial. The platform should monitor the network conditions, the time and location of the experiment, as well as the metadata from the modems, including, for example, cell ID, signal strength and connection mode.

Easy to use platform: It is crucial to make it easy for users to access the system and deploy experiments on all or a selected subset of nodes. This requires a user friendly interface together with a well managed and fair scheduling system.

2.2.2 Design Overview

We shaped the main building blocks of the MONROE platform such that we can meet the above-mentioned requirements. Note that while implementing different components of the platform, operational aspects also impacted the design choices, which we will discuss in detail in Sections 2.4 and 2.5. Next, we give an overview of the purpose and functionality of the main building blocks of the MONROE system, which we illustrate in Figure 2.1. All the software components of the MONROE system are open source [28].

MONROE Node: MONROE operates 150 nodes in 4 countries in Europe (Spain, Italy, Sweden, and Norway). The measurement node resides at the core of our platform. Its design comprises two main notions, namely the hardware configuration, and the software ecosystem. In terms of hardware, each node has a main board that

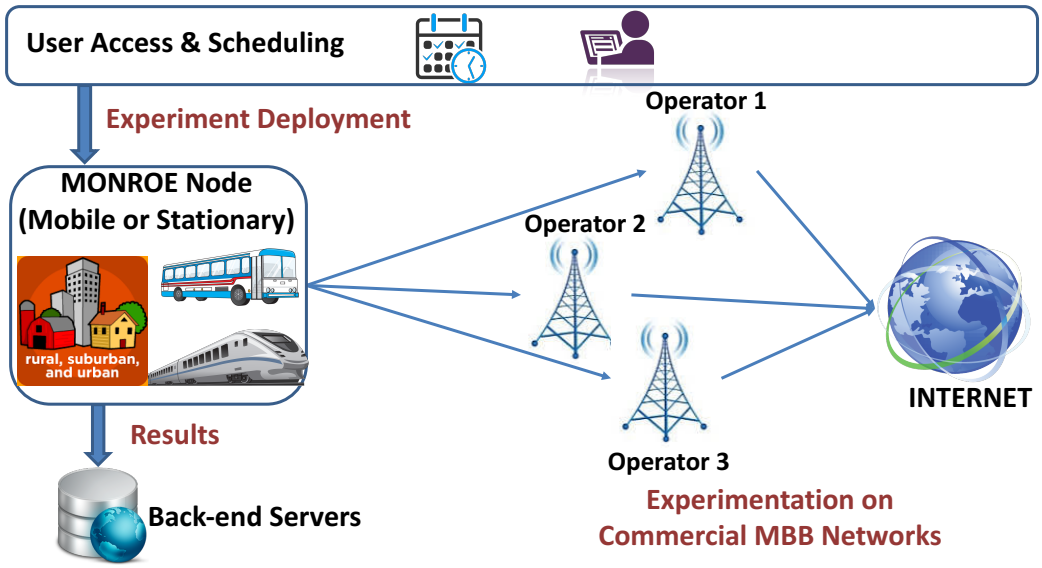


Fig. 2.1 The MONROE platform: MONROE Nodes operate in trains, buses or inside homes and each connects to three commercial mobile operators in each country with MONROE presence. Users access the available resources and deploy their experiments via the User Access and Scheduling. Measurement results synchronize to external repositories operating in the back-end.

is a small programmable computer and supports (at least) 4 interfaces: three 3G/4G modems and one WiFi modem. To cover a diverse set of mobility scenarios, we customize a portion of the nodes (i.e., 95 out of 150 total nodes) to operate on public transport vehicles (buses and trains) and also in delivery trucks. In Section 2.3, we detail the choices for the node hardware implementation and our experience with running two node prototypes.

The node software is based on a Linux Debian ”stretch” distribution[2] to ensure compatibility with multiple hardware configurations and to enable a large set of experiments. Furthermore, especially considering the experimentation on protocols, Linux is the only operating system with sufficient hardware support for research and implementation of transport protocols due to the accessibility of the source code, flexibility and community maintenance to ensure operability with other systems. On top of the operating system, the nodes run: (i) *the management software* that performs the normal jobs expected on any mobile device, (ii) *the maintenance software* that monitors the operational status of the nodes and diminishes manual maintenance intervention, and (iii) *the experimentation enablers* that enable experi-

[2] https://wiki.debian.org/DebianStretch

ment deployment (via the scheduler client) and feed rich context information to the experiments. To provide agile reconfiguration and access for the experimenter to different software components, the experiments run in the Docker [29] light-weight virtualized environment. This also ensures containment of external actions in the node system. We periodically transfer the results of the experiments from the nodes to a remote repository. We further detail in Section 2.4 the node software ecosystem and present our evaluation of potential node internal performance overheads.

User access and scheduling: MONROE enables access to platform resources through a user-friendly web portal [30] that allows authenticated users to use the MONROE scheduler to deploy their experiments. The MONROE Scheduler facilitates exclusive access to the nodes (i.e., no two experiments run on the node at the same time) while ensuring fairness among users by accounting data quotas. We provide the details and the implementation choices for the user access and scheduling policies in Section 2.5.

2.3 Hardware Implementation

Given the requirements we drew from MONROE stakeholders (Section 2.2), the measurement device needs to be small, able to function in different environments (buses, trains, and homes), affordable, robust, sufficiently powerful, and should support the mainline Linux kernel. The size and price constraints limited us to evaluate different Single Board Computers (SBCs). There is a large amount of different SBCs available to the consumer public, with different CPU architectures and hardware configurations. However, most contain hardware requiring the use of proprietary drivers, thus restricting us to old kernels or making it impossible to compile custom kernels. We evaluated several options, including popular ones such as Raspberry Pi [31], Odroid [32], Beaglebone [33] and we selected PC Engines APU [34]. We chose the APU because it provides sufficient processing power, storage and memory for the foreseeable future at a reasonable cost. APUs integrate a 1Ghz 64 bit quad core processor, 4GB of RAM and a 16GB HDD. APUs have 3 miniPCI express slots, two of which support 3G/4G modems.

Modem Selection: To multihome to three mobile operators and a WiFi hotspot, we initially equipped the PC Engines APU board with an Yepkit self-powered USB hub [35], three USB-based CAT4 MF910 MiFis [36] and one WiFi card [37]. The

reason we chose the MF910 MiFi is because, at the time we selected the hardware, it was the most modern device sold by operators we measured.

In the prototype validation phase, this implementation presented some major obstacles. While the APUs proved to be very stable, the MiFis proved more challenging than expected. First of all, in the last quarter of 2016, the MiFis' vendor issued a forced update to the firmware. The update was applied despite the fact that we took special care to configure the devices not to receive automatic updates. As a result of the forced update, all our MiFis became inaccessible for the MONROE system. Furthermore, the MiFis themselves were prone to resets or to enter a working state (transparent PPP) from which we could only restore them to normal operation by draining their batteries, or performing a manual reboot by pushing the power button. Finally, after 6 months of operation, some of the MiFis showed clear signs of swollen batteries. This problem brought serious safety concerns for the nodes operating in places other than our own (controlled) premises (e.g., public transport vehicles). We thus modified the hardware configuration to use internal modems operating in the miniPCIe slots of the APU board.

Current Node Configuration: We decided to increase the control over the MONROE node and base its implementation on a dual-APU system. One of the two APUs in each node has two MC7455 miniPCI express (USB 3.0) modems [38], while the other has one MC7455 modem and a WiFi card. We chose Sierra Wireless MC7455 [3] as our 4G modem since, at the time of the upgrade, it was supporting the most recent category (CAT6) an industrial grade modem could provide. This design eliminates the risk brought on by the use of batteries, avoids any forced updates (the new modems are not routers), simplifies resets (no draining of battery) and increases our overall control over the system.

Takeaways: APUs showed very stable performance, while re-purposing the MiFis to behave as simple modems presented major challenges (e.g., forced updates and swollen battery problems). We thus bring forward a more compact and robust node configuration that relies on internal modems operating in miniPCIe slots. This also simplifies the node since we avoid potential NAT and routing issues the MiFis might trigger.

[3] https://source.sierrawireless.com/devices/mc-series/mc7455/

2.4 Node Software Implementation

In this section, we describe in detail the node software ecosystem and present the justification for our implementation choices.

2.4.1 Software Ecosystem

Fig. 2.2 presents the elements that coexist in the MONROE node software ecosystem, namely the node management software, the node maintenance software and the experimentation enablers.

The **node management software** integrates a set of core components that run continuously in the background. They perform low-level work in line with the normal jobs expected on any mobile device or computer. These include (i) a *Device Listener*, which detects, configures and connects network devices, (ii) a *Routing Daemon*, which acquires an IP address through DHCP, sets up routing tables, and (iii) a *Network Monitor*, which monitors interface state, checks the connectivity of the different interfaces and configures default routes. The node operates behind a firewall, which we configure with strict rules to increase node security.

The **node maintenance software** integrates components that monitor the node status and trigger actions to repair or reinstall when malfunctioning. A *system-wide watchdog* ensures that all core components (node management) are running. However, during the first few months, we experienced loss of connection to nodes because of problems that watchdogs could not tackle, such as file system corruptions which can occur due to frequent sudden power loss in mobile nodes. Thus, we defined and implemented a robust node recovery method, called BootOS, that enables a hard restart of the node (i.e., a reinstallation of the operating system to a known working baseline). This method allows us to recover both from file system errors that prevent system boot-ups, and software configurations that may lead to loss of connectivity. To achieve this goal, we trigger a two-stage boot loader process at node start-up. In the first stage, we start the BootOS, which resides entirely in RAM and only uses read-only hard-drive access for its normal operation. The BootOS verifies that the filesystem of the APU is not corrupt, and that no forced reinstallation has been requested. It then proceeds to boot the MainOS, which contains the MONROE

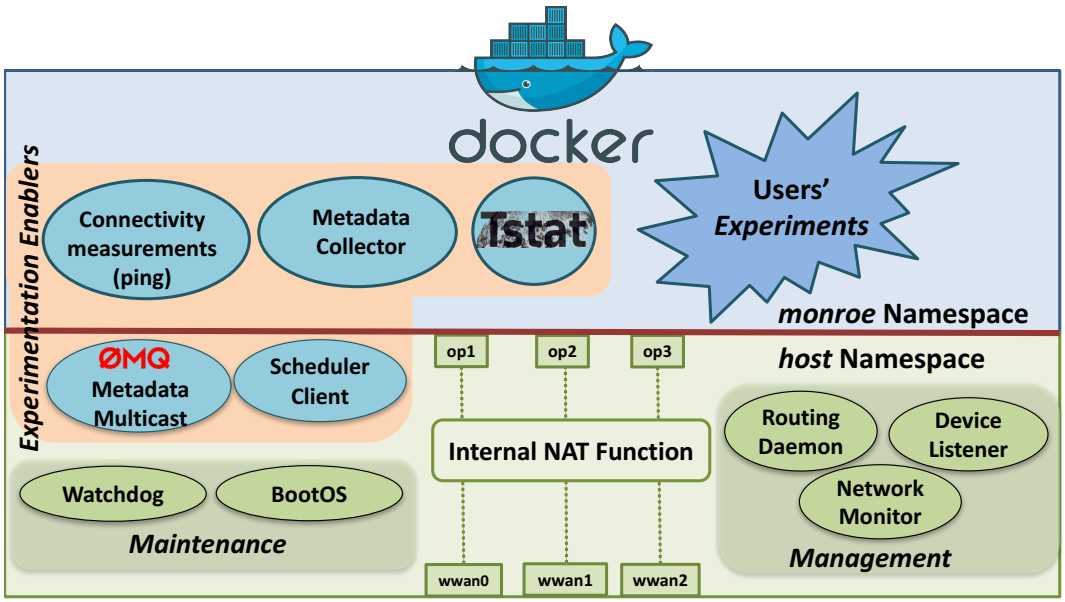


Fig. 2.2 Node Software Ecosystem

system software. If the filesystem is corrupted, or in case of a forced reinstallation, the BootOS reinstalls an image of a known working installation.

The **experimentation enablers** include the scheduling client, the default experiments, and the services for external experiments. Within the node software ecosystem, we differentiate between the user experiments and the management and maintenance software by configuring a separate *MONROE network namespace* where experiments run. This increases our control over the ecosystem and limits the impact external users can have on the node. This separation further allows us to account (as part of the scheduling system) the traffic volume each user consumes. We require that each experiment runs inside a virtualized environment (Docker container) to ensure separation and containment of processes. The *Scheduling Client* communicates with the Scheduler to enable experiment deployment per user request. It periodically checks for new experiment containers to run in the node and deploys them in advance to their scheduled execution time. Section 2.5 provides more details on the scheduling system. The *metadata broadcasting service* runs continuously in the background and relays metadata through ZeroMQ [39] in JSON [40] format to experiment containers. The nodes periodically run connectivity measurements (e.g., ping), and this together with metadata allow us to monitor the node's state and the overall health of the platform. Furthermore, the *Tstat* [16] passive probe provides

insights on the traffic patterns at both the network and the transport levels, offering additional information on the traffic each interface exchanged during an experiment.

Takeaways: Containment of users activity in the node is paramount to avoid security risks, node malfunctioning events, unreliable results and, more severely, node loss. We prevent foreign unauthorized access to the node with a strict firewall. Then, continuous monitoring of the platform is crucial and we enable it by implementing monitoring functions in the node management software. Node maintenance is expensive, so it is important to forge the node as a self-healing system. We implement this functionality in the node maintenance software that takes automatic actions when the node malfunctions.

2.4.2 Experiment Containment

Docker Virtualization: The node design we propose mandates that MONROE users execute their experiments inside Docker containers, which provide isolation from the host node. This is true both for default monitoring measurements and external users experiments. Docker containers are based on a layered file system, where a container can reuse layers shared with other containers.

MONROE provides the default base image for the experiment containers, which integrates the base operating system installation with default tools that are potentially useful for many experiments. The lightweight containers provide just the contents that are unique for the particular experiment, significantly reducing the download and deployment time overhead and accountable traffic volume. Running experiments inside a container have access to the experimental network interfaces. They can read and write on their own file system, overlaid over that of the base MONROE image. Finally, there are specific paths (e.g., `/MONROE/results/`) where the experiments can write their results and that the node automatically transfers to the MONROE servers. Our public software repositories contain all the files necessary to build new user experiments, as well as experiment templates and examples.

Internal NAT Function: To ensure the minimum impact of user experiments gone wrong, we define the *MONROE network namespace* where experiment containers run. For each physical interface that the network-listener detects as available, we create a virtualized Ethernet, `veth`, interface pair, and move one end to the MONROE namespace. We then add routing rules in the network namespace to allow routing

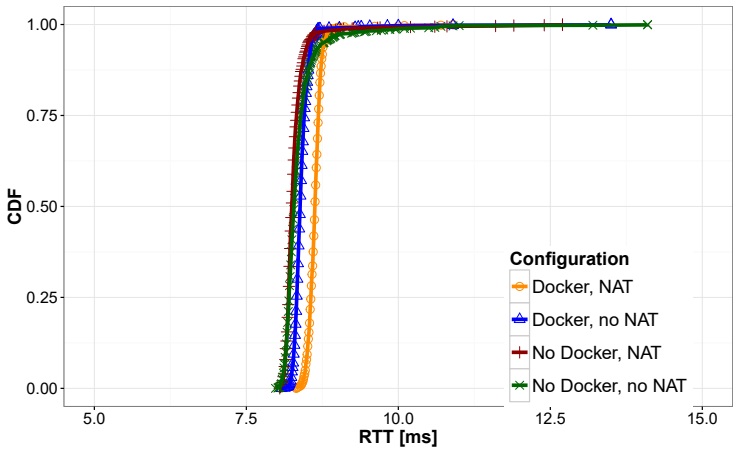


Fig. 2.3 CDFs of ICMP RTTs [ms] measured against 8.8.8.8 per testing configuration over Fast Ethernet link

by interface. In order to allow the network devices in the host namespace to communicate with the ones in the MONROE network namespace, we define an internal NAT function. We use `iptables` NAT masquerading rules in the host namespace to configure the NAT function. Finally, we add the corresponding routing rules to map each `veth` interface to the correct physical interface.

Overhead Quantification: The internal network design introduces two potential overheads that might impact performance measurements: (i) the internal NAT function that connects the network devices in the host namespace with their corresponding duplicates in the monroe namespace, and (ii) the Docker containers we use to separate the processes that correspond to a certain experiment that runs inside the container. Thus, prior to detailing the measurement results of different commercial MBB operators, we focus here on these two design overheads and aim to quantify their impact (if any) on performance measurement results. Specifically, we quantify the delay overhead by running ICMP ping measurements, and the impact on throughput by running HTTP downloads.

To instrument our system benchmarking measurements we use a single APU node running the Debian "stretch" MONROE image with a local Fast Ethernet link. Using a local link allows us to minimize the impact of the network on our measurements, and focus on the impact of the system overheads. We run http download measurements with `curl` and ICMP ping measurements with `fping` to quantify the impact of the internal NAT function and of the Docker virtualization. We focus on four configurations for our testing setup, namely: no NAT and no Docker

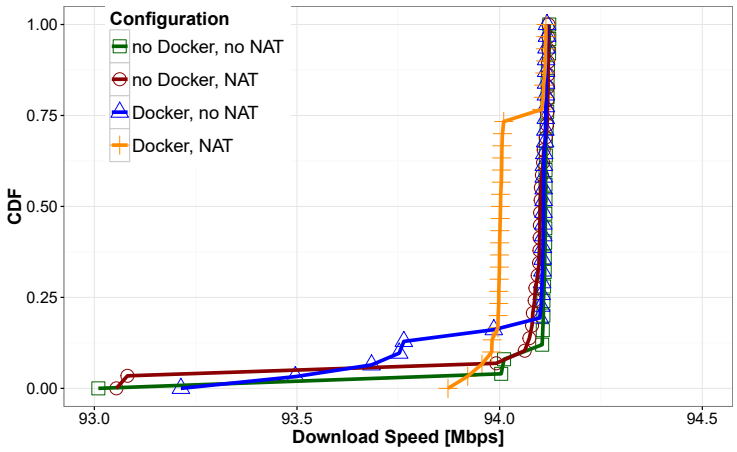


Fig. 2.4 CDFs of Downloads Speed [Mbps] measured per testing configuration over Fast Ethernet link

(experiments run in host namespace), no NAT but Docker (experiments run inside a Docker container in the host namespace), internal NAT and no Docker (experiments run in the MONROE namespace) and internal NAT and Docker (experiments run inside a Docker container in the MONROE namespace).

To quantify the delay overhead, we collect 1,000 RTT samples against the Google DNS server 8.8.8.8 on the Ethernet connection on all four configurations. Fig. 2.3 shows the results of the measurements. We conclude that the overhead of the NAT function internal to the node is insignificant. In average, we see a penalty in the order of 0.1ms, (i.e., in the range of clock granularity in Linux systems). We note that the Docker and NAT combination introduces a slight delay, which is not overwhelming.

For the throughput measurements, we download 1GB of data from a server configured in the local network. We collect 30 samples for each testing configuration. In Fig. 2.4, we show the Cumulative Distribution Function (CDF) of download speed per namespace and operator, for each of the different targets. We find that there is a 1% performance penalty that using the internal NAT function and the Docker virtualization introduces in average. We report no direct impact of using the Docker containers, which we expected, since the purpose of the Docker virtualization is purely for experiment containment.

Takeaways: Our priority in the node software implementation phase is keeping the nodes within normal functioning parameters for as long as possible and limiting direct maintenance intervention, while allowing external users to run a wide range of

complex measurements with minimum interference. To achieve this, we separate the network namespace where users can run their experiments from the host namespace, where the monitoring and management software runs. This introduces two potential overheads in the system, which we quantify and show to have little or no impact.

2.5 User Access and Scheduling

We provide access to the MONROE platform through a user-friendly interface consisting of an AngularJS-based web portal [30]. As part of the MONROE federation with the Fed4FIRE [41] initiative, the user access follows the Fed4FIRE specifications in terms of authentication and resource provisioning. Through the portal, experimenters interact with the scheduler and deploy their experiments without accessing directly the nodes. The scheduler API is accessible to enable experiment deployment automation. The scheduler prevents conflicts between experiments (i.e., only one user can run an experiment on a certain node at a time) and assigns resources to each user based on their requirements and resource availability.

Given the challenging scenarios we aim to cover in our testbed, nodes in MONROE have potentially unreliable connectivity and low bandwidth. This is the norm for node in buses, trains, and trucks, which follow the schedule of the host vehicle. Experiment scheduling therefore accounts for two factors: (i) the node may not have connectivity at the time of the experiment, and (ii) a high lead time when deploying containers means that experiments should be deployed early. Furthermore, experimenters may require to run synchronous measurements on multiple nodes. The common approach to task scheduling and decentralized computing, which deploys jobs to registered nodes based on their availability, struggles with these constraints. Therefore, for the MONROE scheduler, we follow a calendar-based approach, assigning time slots to experiments. Deployment of experiments takes place up to 24 hours in advance, as soon as the node retrieves information about the assigned task. It allows both immediate scheduling on nodes that are not otherwise occupied, and scheduling synchronous experiments on low availability nodes well in advance. It also allows synchronizing experiment runtime with vehicle schedules when available.

In addition to manage the time resource, the scheduler handles data quotas assigned by the contracts with the MBB operators. We assign each experimenter a fix data quota. In addition, we may assign users a quota on computing time (i.e.,

maximum time the users can run experiments on the node). We designed the quota system to provide fair usage of the available resources. An important factor to ensure fairness in day-to-day usage, is that a certain data quota is reserved by the experimenter in advance, and subtracted from the user quota for the duration of the experiment. Experimenters may subsequently refund the remaining quota. Hence, it is not possible to block large quantities of resources without having been assigned the necessary budget, even if the resources are not actually used.

From March 2016 until March 2017, the MONROE scheduler has been actively used by 30 users. A total of 75,002 experiments have successfully ran on the platform, while 7,972 scheduled experiments failed. There are many different reasons for failed experiments, for example that the container exits unexpectedly or the data quota is exceeded. Note that these failures are expected especially for the new users that are trying to familiarize themselves with the platform. We are running an open conversation with our users, gathering feedback from them and updating the user access and scheduling policies accordingly.

Takeaways: Resource allocation and experiment scheduling on MONROE are challenging because nodes have potentially unreliable connectivity (e.g., nodes in mobility scenarios) and limited data quota due to commercial-grade subscriptions. A calendar-based approach for scheduling addresses these requirements by taking into account per user and per node data quota, and synchronized experiment start time.

2.6 Open Experimentation

Starting from the platform design phase, we have been working together with our stakeholders to understand their requirements from the MONROE system and which experiments have the highest appeal (Section 2.2).

2.6.1 MONROE Experiments

We present the base experiments deployed by the consortium. We are currently offering to the community a series of experiments [42], which any external users can deploy on their own. This goes toward achieving our goal of shaping MONROE into an Experimentation as a Service (EaaS) platform. We group all these experiments in

three main categories: MBB Performance, Service Oriented QoE, and Innovative Protocols and Services. These categories also fit to the range of measurements that our users are currently curating and have been already actively deploying. The distribution of experiment runs on the MONROE platform to the time of writing among these categories is: MBB Performance (19%), Service Oriented QoE (36%) and Innovative Protocols and Services (45%). The volume of data that experiments in different categories consume varies, with Service Oriented QoE taking the largest quota (60%), while Innovative Protocols and Services are the least demanding (10%), despite registering the largest number of experiment runs. We further detail each category and provide examples of experiments and analysis one can perform using MONROE.

MBB Performance

To measure a mobile network in a reliable and fair way, it is important to identify the metrics that accurately capture its performance. Different stakeholders have different metrics of interest and we argue that MONROE is able to cater all of them (Chapter 3).

Service Oriented QoE

An important measurement dimension to explore comes from the great interest in how users perceive individual services and applications over different terminals (e.g., mobile phones, tablets, and computers). The recent proliferation of user-centric measurement tools (such as Netalyzr [7]) to complement available network centric measurements validates the increasing interest in integrating the end user layer in network performance optimization. MONROE enables experimentation with essential services and applications, including video streaming, web browsing, real-time voice and video, and file transfer services (Chapter 4).

Innovative Protocols and Services

Another significant use case for MONROE is investigating the impact of middleboxes in the current Internet ecosystem. These range from NATs to security devices to performance enhancing TCP proxies. Middleboxes are known to introduce a series

of issues and hinder the evolution of protocols such as TCP. Since middleboxes are ubiquitous in MBB networks [43–45], in collaboration with the H2020 MAMI project [46] we aim to observe and characterize middlebox operations in the context of real-world MBB deployments (Chapter 3). MONROE further enables assessment of new protocol innovation (Chapter 6).

2.7 Conclusions

In this chapter, we reported on our experience designing an open large-scale measurement platform for experimentation with commercial MBB networks. MONROE is a completely open system allowing authenticated users to deploy their own custom experiments and conduct their research in the wild. The platform is crucial to understand, validate and ultimately improve how current operational MBB networks perform towards providing guidelines to the design of future 5G architectures. We described our experience with the MONROE system implementation and detailed the hardware selection for the MONROE measurement node, its software ecosystem and the user access and scheduling solution. We emphasized the versatility of the design we propose, both for the overall platform and, more specifically, for the measurement nodes. In fact, the node software design is compatible with a number of different hardware implementations, given that it can run on any Linux-compatible multihomed system. Our current hardware solution is the most fitting for the set of requirements and the predicted usage of MONROE, which we evaluated based on our discussions and interaction with the platform's users.

Chapter 3

Speedtest-like Measurements in MBB Networks

3.1 Introduction

The society's increased reliance on MBB networks has made provisioning ubiquitous coverage and providing high network performance and user QoE the highest priority goal for mobile network operators. This motivates researchers and engineers to further enhance the capabilities of MBB networks, by designing new technologies to cater for a plethora of new applications and services, for the growth in traffic volume, and for a wide variety of user devices.

When coming to performance assessment, the picture is much more complicated in MBB networks than in wired networks. Even the simplest of the tests, i.e., a "speedtest-like" measurement of the single TCP bulk download speed using HTTP, may become complicated to interpret in MBB networks, due to the large number of factors that affect performance. Physical impairments, mobility, variety of devices, presence of Performance Enhancing Proxies (PEP), different access network configurations, etc., all possibly impact the measurement results, and complicate the picture.

When facing performance assessments, a common approach is to rely on end users, and their devices, to run tests by visiting a website [6], or running a special application [7]. Federal Communications Commission (FCC) follows a similar crowdsourcing approach to measure MBB networks in the USA [27]. Network oper-

ators and independent agencies sometimes perform drive tests to identify coverage holes or performance problems. These tests are, however, expensive, do not scale well [47], and little information on methodology is given.

Here, we rely on the MONROE (Chapter 2) open platform, that offers an independent, multihomed, large-scale monitoring platform for MBB testing in Europe. Despite the large dataset, and the scientific approach, we find that running even a simple speedtest-like experiment proves to be very complicated, with results that apparently vary on a large scale, with no obvious correlations, and sometimes in an unpredictable way. We observe the presence of NAT, and of transparent proxies, as well as different access network configurations, and roaming agreements, each adding complexity to the already complicated picture. Thanks to the MONROE platform, we design and run further experiments to corroborate our findings, and better understand the results.

While preliminary, we present our finding (and make available all raw data) in the hope to shed some light into the debate about performance assessment in MBB environments. Indeed, since the issue is far from trivial, we believe there is a need to define benchmarking principles that allow to fairly compare performance in MBB (and soon in 5G) networks.

The rest of this chapter is organized as follows. In Section 3.2, we present the motivation of this work. In Section 3.3, we briefly discuss the related work. In Section 3.4, we describe the measurement approach we use to collect and analyze the collected dataset. Our methodology is discussed in Section 3.5. In Section 3.6, we present our finding. Finally, in Section 3.7, we conclude the chapter and we discuss future research issues.

3.2 Motivation

To take a first look into speedtest measurements in commercial MBB networks, we conducted an initial measurement campaign, and measured different speedtest apps under the same conditions, using an Android phone as a regular user could do, from home. There are a number of crowdsourced apps for measuring MBB performance via end-user devices. Among them, we choose the most popular ones: *Speedtest* by

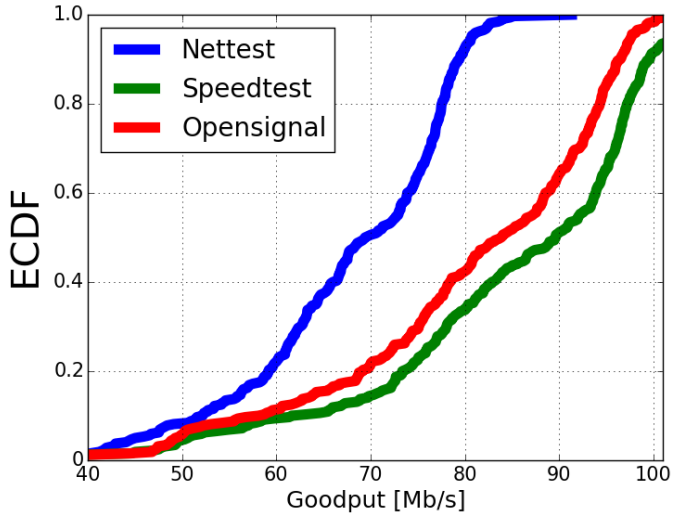


Fig. 3.1 ECDF of reported download rate for different tools in 4G

Ookla [6], *OpenSignal* by OpenSignal [48], *RTR-Nettest* by Austrian Regulatory Authority for Broadcasting and Telecommunications (RTR) [49].

Typical performance measurements by such tools comprise Downlink (DL) and Uplink (UL) data rate, and latency. Here we focus on download speed only. For our measurement campaign, we run speedtest measurements with Speedtest (v3.2.29), OpenSignal (v5.10), and Nettest (v2.2.9). To ensure the fair comparison of the tools, we run the tools in rounds where each tool is run one after the other and in randomized order on a stationary measurement device located in Oslo, Norway, when connected to the same network in 4G.

We ran 320 batches of measurements in total. Fig. 3.1 shows the Empirical Cumulative Distribution Function (ECDF) of download rate values reported by the tools. Surprisingly, we observe a large variation in measurements, both within runs of the same tool (max-min variation of 60 Mb/s, see the Opensignal in Fig. 3.1), and between tools (max-max variation of 20 Mb/s range, see the difference between Nettest and Speedtest in Fig. 3.1).

These large differences indicate a significant variation in both measurement methodology and network condition, which we have confirmed through the reverse-analysis of traffic traces collected during measurements with different tools. Thus the natural question is "Can we reliably benchmark download speed in MBB networks?".

3.3 Related Work

The analysis of MBB network performance, and its prediction are on the research agenda of the networking community. There are mainly three approaches for measuring the performance of MBB networks: (i) crowd-sourced results from a large number of MBB users [50, 51], (ii) measurements based on network-side data such as [52–54], and (iii) measurements collected using a dedicated infrastructure [55–57]. Network-side and active tests can be combined in the so-called "hybrid measurements" approach, as implemented, e.g., in [58]. In this thesis, we collect data from a dedicated infrastructure in order to have full control over the measurement nodes, allowing us to systematically collect a rich and high quality dataset over a long period of time.

In the literature, some studies take it one step further and focus on the mobile infrastructure (e.g., presence of middleboxes) and its impact on performance. Performance enhancing middleboxes are widely deployed in the Internet and it is of great interest to measure and characterize the behavior of them especially in MBB networks where the resources are scarce. The impact of middleboxes on measurements was explored in [59] where the authors proposed a methodology for measurements in MBB networks. Farkas et al. [60] used numerical simulations to quantify the performance improvements of proxies in LTE networks. In [54], the authors analyzed LTE data collected in one city, to study the impact of protocol and application behaviors on network performance, mostly focusing on the utilization of TCP. Becker et al. [61] worked on analysis of application-level performance of LTE, and detected middle-boxes deployed on LTE networks, studying their impact on the measured performance. The most thorough analysis to characterize the behavior and performance impact of deployed proxies on MBB networks was carried out in [45] where the authors enumerate the detailed TCP-level behavior of MBB proxies for various network conditions and Web workloads. Although the common belief is that proxies provide performance benefits, Hui et al. [62] showed that they can actually hurt performance by revealing that direct server-client connections have lower retransmission rates and higher throughput. Wang et al. [44] showed how MBB middlebox settings can impact mobile device energy usage and how middleboxes can be used to attack or deny service to mobile devices. Taking a different route, Kaup et al. [63] studied the root causes of MBB network performance variability by means of measurements in one country, and showed that management and configuration

Table 3.1 The number of experiments in the dataset

country	City (sites)	Operator	# Nodes	# Experiments
Italy	Torino(4) Pisa(5)	op0	12	1995
		op1	14	2184
		op2	14	2316
Sweden	Karlstad(7)	op0	28	3029
		op1	28	2644
		op2	28	3117
Spain	Madrid(6) Leganes(5)	op0	18	4924
		op1	15	3502
		op2	7	1888
Norway	Fornebu(3) Oslo(4) Bergen(4)	op0	13	2437
		op1	12	2220
Total	8	11	73	30256

decisions have a considerable impact on performance. We differentiate our work from these studies by focusing on different countries and operators. Furthermore, these studies consider a snapshot of the experiments which bound results to the measured ISP network and to the geographical location of the setup. On the contrary, our approach and experiments, by using the MONROE platform, allowed us to collect data through continuous experiments over 4 countries and 11 operators. Our goal is to understand the mobile ecosystem and whether a simple speedtest can be run reliably over the current complex mobile networks, rather than measuring the performance of the mobile networks or the impact of middleboxes.

In closing, we remark that even performance measurements in wired networks can be a fairly complex task, because of user preferences, of the influence of users' home networks, of ISP traffic shaping policies, as noted by Sundaresan et al. in [64], who studied the performance of wired networks observed from home gateway devices, and observed counter-intuitive results.

3.4 Measurement Setup

In this section, we briefly describe the collected dataset.

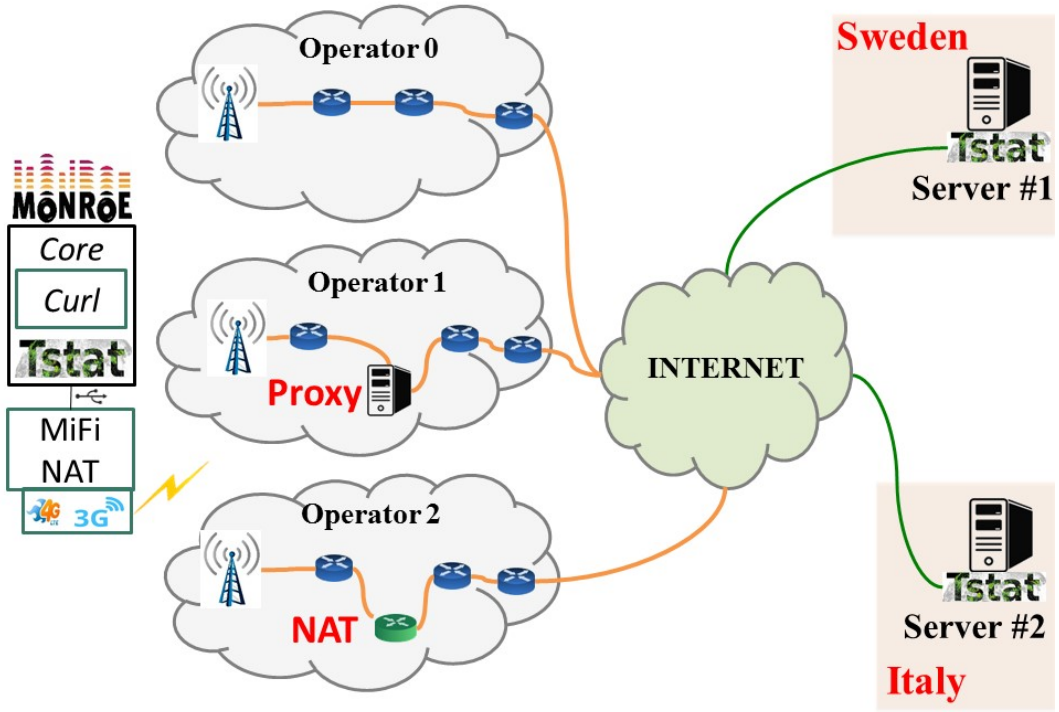


Fig. 3.2 Experiment setup

3.4.1 Basic HTTP Test

Fig. 5.3 shows the experiment setup we consider in this chapter. The leftmost element is the MONROE node. It contains the core components, with containers that run active experiments. Traffic generated by the applications passes through the selected MiFi modem (initial modem configuration Section 2.3) where a NAT is in place, then goes through the ISP network, and the Internet, toward the selected server – on the rightmost part of figure. Each node runs also Tstat [16], a specialized passive sniffer. Tstat captures traffic on each MBB interface and extracts statistics by passively observing packets exchanged with the network. Another instance of Tstat runs on the server side, thus capturing and processing traffic at the other end of the path.

As previously mentioned, each MONROE node regularly runs a basic set of experiments. Among these, the HTTP download experiment uses single thread *curl* to download a 40 MB file for a maximum of 10 seconds from dedicated and not-congested servers in two countries, one in Italy, one in Sweden[1]. Network configuration may change from country to country, and from operator to operator as depicted in Fig. 5.3. Beside the NAT at the MiFi router, the ISP can provide a public IP address to the modem (e.g., Operator 0) and no other NAT or middlebox on the path. Alternatively, the ISP might use some kind of PEP (e.g., Operator 1), or it can use Carrier Grade NAT to do NAT/NAPT (e.g., Operator 2).

In this chapter, we consider measurements that were run during September and October 2016 in four countries and different sites. We consider only stationary nodes. The experiment ran every 3 hours in synchronized fashion. Table 3.1 reports the total number of nodes and the number of experiments for each operator. Overall, we collected more than 30 000 experiments from 11 operators. ISPs were subjected to different numbers of experiments. The reason can be coverage holes, exhausted data quota on subscriptions, or rare failures inside the nodes. The name of the ISP is specified by a number, to avoid exposing the operator name – our goal is not to provide a ranking among ISPs but rather to observe if it would be possible to reliably measure performance. During experiments, all networks were in normal operating conditions (and unaware of our tests).

The active application and passive flow-level traces on the client and server sides cannot give us information about the technology and signal strength at the MBB channel during the experiment. Therefore, we use the metadata collected by the MONROE platform to augment the information about the access link status. The MONROE metadata are event-based data collected by passively monitoring the statistics exposed directly from the MiFi modems through their management interface. This data is transmitted and stored in the project database for analysis, and can be easily correlated to each node and interface.

[1] During the HTTP test no other experiment can run. The 3 h periodicity and 10 s limit are imposed to avoid booking the platform for long time. The 40 MB file size limits the total volume of data to less than 9.6 GB/month and avoids to erode the limited data quota of each subscription.

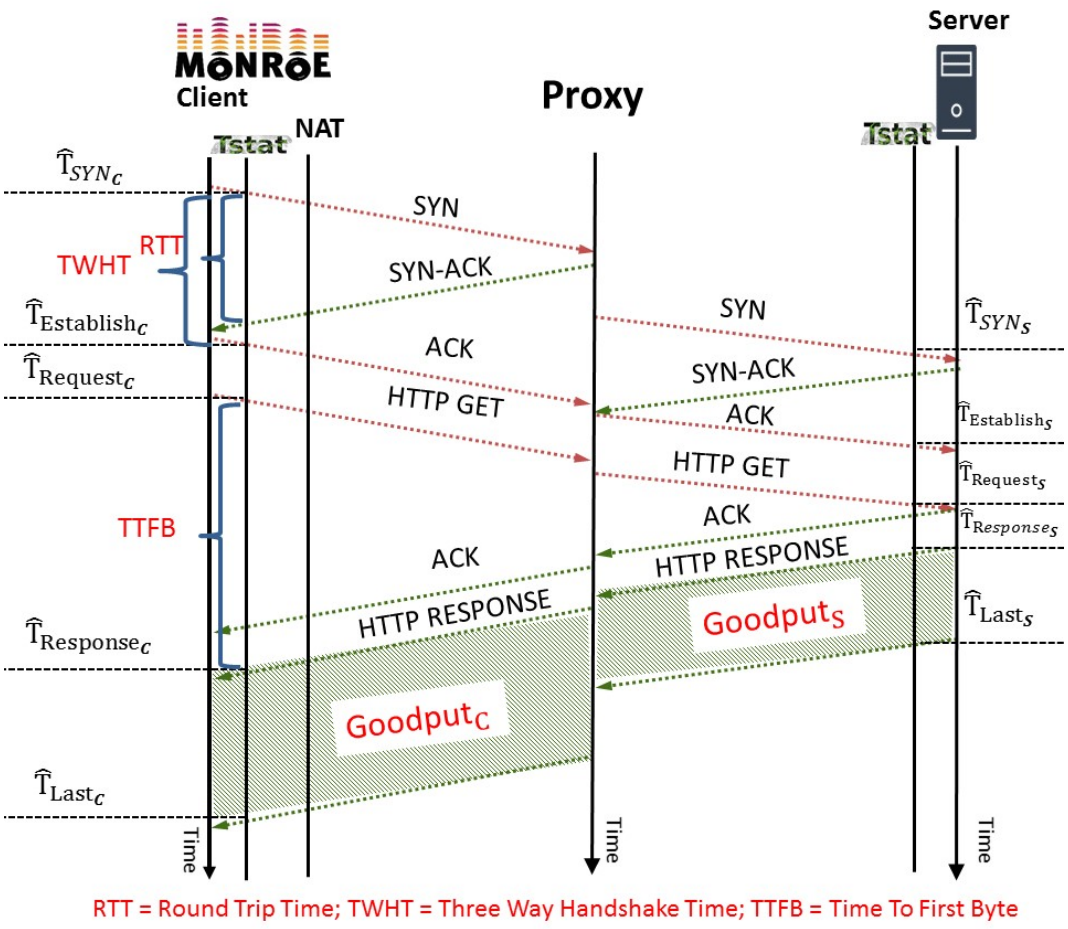


Fig. 3.3 Packet timeline in case of PEP in the path

3.4.2 Additional Tests

To verify some of the hypotheses about the presence of NAT or PEP in the ISP network, we additionally instrumented a subset of nodes to run HTTP tests, but against HTTP servers running on different TCP ports. In particular, we checked possible HTTP-related ports (80, 8080), HTTPS port (443), and random ports (4981, 19563). Again, Tstat runs on both client and server, and lets us verify the presence of middle-boxes by contrasting the measurements on both sides.

3.5 Methodology

Here we detail the methodology we used to process the collected data. Let us first start describing in more details the available information at our disposal.

3.5.1 Measurement Definition

Fig. 3.3 reports the possible setup during an experiment. The client (on the left) opens a TCP connection, and fetches the file via HTTP. Tstat on the client side sniffs packets, and extracts measurements by correlating the sent and received segments. For instance, it extracts the Round Trip Time (RTT) of each TCP segment/acknowledgement pair, the Time to complete the Three Way Handshake Time (TWHT), the Time To receive the First Byte from the server (TTFB), and the download speed. In the example, there is a PEP, which terminates the TCP connection from the client side, while opening another one toward the server. The second Tstat instance running on the server observes the segments being exchanged between the PEP and the server, and collects statistics that we can later contrast with those collected on the client side.

We now define the most important measurements we use in this work. We indicate measurements collected on the client side or server side with subscript C or S, respectively.

Goodput – $\widehat{G}$

$\widehat{G}$ is the most important measurement, and is defined as the average rate at which the client receives information at the application layer. Let $\hat{T}_{Response_C}$ and $\hat{T}_{Last_C}$ (see Fig. 3.3) be the timestamps of the first and the last data packet at the client side, and let D be the size of the application payload size sent by the server. We define the client-side goodput as:

$$\widehat{G}_C = \frac{D}{\hat{T}_{Last_C} - \hat{T}_{Response_C}}$$

Since Tstat is co-located at the client, this measurement is actually the same as the measure computed directly by the *curl* application.

Round Trip Time – RTT

Tstat measures the RTT by matching the data segment and the corresponding acknowledgement in a flow (as depicted in Fig. 3.3). For each segment/ack pair, Tstat obtains a RTT sample. It then computes the average, standard deviation, minimum and maximum among all RTT samples seen in the same TCP connection. On the client side, Tstat gets a reliable measurement of the RTT between the TCP client and the TCP server (or PEP) nodes. On the HTTP server, Tstat measures the RTT from the server to the client (or PEP).

Time To Live – TTL

For each packet, Tstat extracts the TTL values from IP packets, and tracks minimum, maximum, and average values seen in all packets of the same TCP flow. On the client side, we consider the maximum TTL observed in packets transmitted by the server (or PEP). This is linked to the number of hops that the packets in the flow have traversed before reaching their destination.

TCP Options

For each TCP connection, Tstat logs information about TCP options such as Timestamps, Maximum Segment Size (MSS) [65], and negotiated window scale factor [66]. In the MONROE platform, all nodes run the same software and hardware. Since we have also control on the server side, we know exactly which options are declared and supported by both endpoints. If the ISP does L4 mangling, or a PEP is present on the path, Tstat could observe different TCP options on the client side and server side.

Received Signal Strength Indicator – $RSSI$

Among the information the MONROE node collects from the modem, we use the RSSI reported in dBm (logarithmic scale) as indicator of the quality of the channel. The RSSI indicates the total received signal power and typically, -100 dBm and -60 dBm indicate low signal level and very strong signal level, respectively. Recall that all nodes use the same MiFi modems, so this information is measured consistently by the platform. We use the RSSI value reported at the time $\hat{T}_{SYN_C}$.

3.5.2 Joining Client with Server Data

All connections go through at least the first NAT at the MONROE node. This implies that Tstat at the client side sees the client *private* IP address provided by the MiFi modem, while Tstat at the server would observe the client *public* IP address.[2] If there is a middle-box in the ISP network, it could further change the IP address, and the port numbers. Thus, matching the connection observed at the server side to the one seen at the client side is not trivial. The MONROE metadata exposes the actual IP address provided by the operator (either private or public) to the MiFi modem, so that we can use this to map connections on the client and server side. We call it "client IP" for simplicity in the following.

Let the client IP provided by operator to the MiFi modem at the node and seen by Tstat at the HTTP server side be indicated by IP_C and IP_S, respectively. Similarly, the client port at the node and HTTP server sides are denoted by $Port_C$ and $Port_S$, respectively.

In case of NAT, NAPT, or in presence of a PEP, $IP_C \neq IP_S$, and it becomes complicated to associate the flows seen in each single experiment (since we lose the information about the originating node). In this case, we associate the flow to the operator by resolving the IP_S address into its owner. We use the *MAXMIND* database [67], and, in case of a miss, we default to *whois* [68].

In more details, we match the *flow* associated with a certain experiment's TCP connection on the node side and HTTP server side if they start within a 1 second time window ($\hat{T}_{SYN_S} - \hat{T}_{SYN_C} < 1$ s), as follows:

1. If $IP_C = IP_S$ and $Port_C = Port_S$, we claim there is no NAT or PEP in the ISP network.

2. If $Port_C = Port_S$, $IP_C \neq IP_S$, and IP_C is a private IP address, we claim there is NAT in the ISP network. We can still associate each single flow by matching $Port_C$ to $Port_S$.

3. If $IP_C \neq IP_S$, $Port_C \neq Port_S$, we claim there is NAPT in the ISP network. We match the operator by looking at the IP_S as above.

[2] The MiFi does not change the TCP port number, but only the client IP address.

Hence, we define a flow at the node and HTTP server sides when the connections start in a 1-second time window, have the same client IP address, the same server port number, and the same client port number (considering the port number is not changed by NAPT or PEP). If this is not possible, we simply assign data collected on the server side to the operator (but we cannot match the single flows). Our analysis shows that the first case can cover most of the operators.

3.5.3 $\widehat{G}$ Mismatch

Given the i-th flow, let $\widehat{G}_C(i)$ and $\widehat{G}_S(i)$ be the goodput recorded by Tstat at the node and HTTP server, respectively. By comparing the observed values, we can show the existence of a PEP in the ISP network:

- $\widehat{G}_C(i) \sim \widehat{G}_S(i)$, illustrates the node experiences almost the same goodput as seen on the HTTP server. In this case, no PEP is present.[3]
- $\widehat{G}_C(i) < \widehat{G}_S(i)$, shows a *mismatch*. In this case, there is a PEP able to download the file from the server with considerably higher $\widehat{G}$ than the capacity on the path from the PEP to the client.

In case we cannot match the single flows, we can still compare statistics of $\{\widehat{G}_C(i)\}$ and $\{\widehat{G}_S(i)\}$ for all flows seen for a given operator.

3.6 Results

In this section we present the results obtained with the experiment setup described in the previous section.

3.6.1 Download Goodput

As a first observation, Fig. 3.4 reports the goodput observed on three of the considered operators during a week, each point presenting the average $\widehat{G}_C$ of a set of experiments

[3] We do not consider exact equality because some packets are in flight, and delay would make $\widehat{G}_S(i) > \widehat{G}_C(i)$ in general.

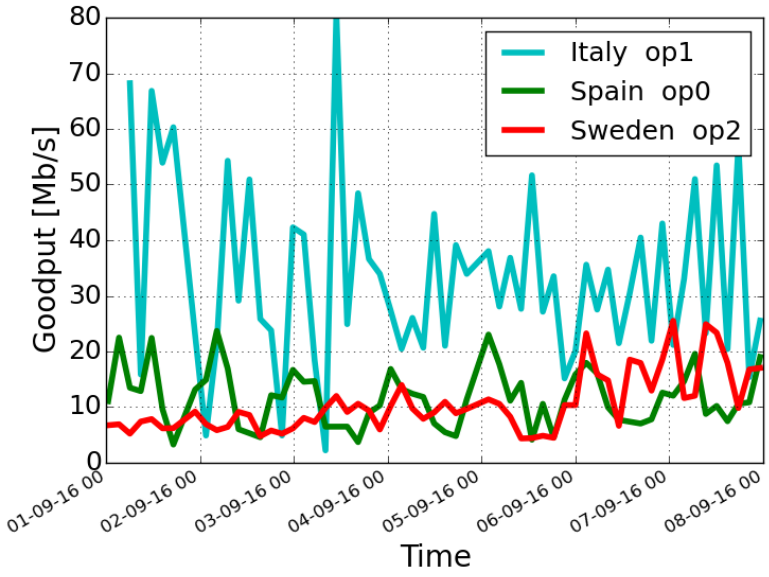


Fig. 3.4 Client-side goodput observed over one week for three operators

in a window of 1000 seconds, i.e., averaging all $\widehat{G_C}$ measurements for that operator during each run every 3 hours. This figure explains the complexity of speedtest-like experiments in MBB networks. Indeed, we observe quite different behaviors, such as i) a daily pattern (op0 in Spain), ii) a change of behavior over time (op2 in Sweden - see the last two days), or iii) unpredictable high variations (op1 in Italy). To check the impact of the duration of the test, and observe the fine grained variability of the capacity measurement, we also report the evolution over time of the download rate measured at the client, every second. Fig. 3.5 shows 2 runs, during which the client downloaded a 1 GB file in no more than 100 s. We observe a large variability, even during a relatively short test. This partly explains the variability observed in Fig. 3.4.

Fig. 3.6 shows the big picture of the client-side goodput observed over the eleven networks we tested in four European countries: Italy, Spain, Sweden, and Norway. Results report the ECDF of the client-side goodput computed from Tstat logs collected in our experiments. The x-axis in each chart of Fig. 3.6 gives the goodput ($\widehat{G_C}$) in Mb/s and the y-axis gives the probability of the goodput being less than the x-axis value. Variability is evident, and confirms the unpredictability seen in Fig. 3.1. Yet, some significant differences exist when comparing operators.

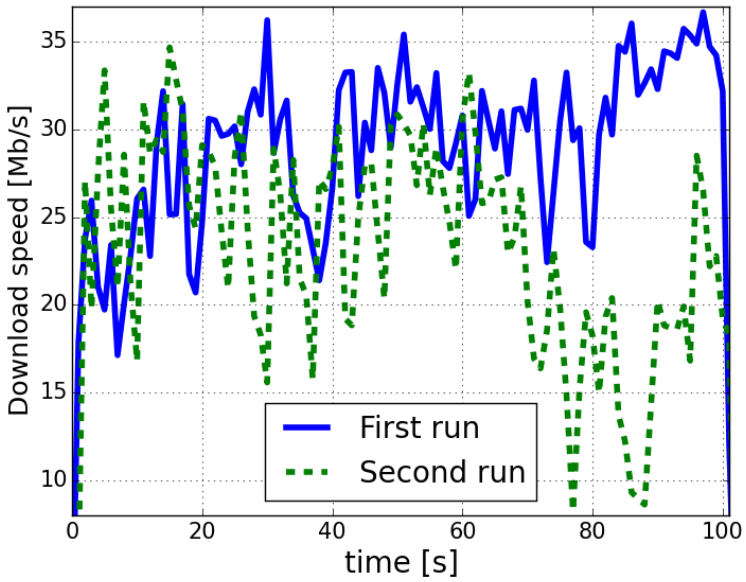


Fig. 3.5 Evolution over time of download speed in two simple run of 100 s on op2 in Italy

In Fig. 3.6d, we see that the two operators we considered in Norway provide similar values of the client-side goodput $\widehat{G}_C$.

On the contrary, the three operators that were measured in Italy gave quite different goodput results. In particular, op0 had a significantly high probability of providing low values of the client-side goodput $\widehat{G}_C$, in comparison to the other two operators. By looking at Fig. 3.8a, that we will discuss in detail later on, the red color of dots of op0 indicate that op0 mostly uses the 3G technology, and is configured so as to have higher RTT with respect to the other two operators. This explains the lower goodput values for op0.

In the case of Spain, we see that op0 in about 40% of the cases provided quite low values of the $\widehat{G}_C$. Our dataset indicates that, during peak times, the goodput provided by this operator is low, as can be seen in Fig. 3.4. We can clearly see that $\widehat{G}_C$ for op0 in Spain exhibits a daily pattern, probably due to throttling in periods of peak traffic. In addition, also by looking at the set of blue squares at the bottom of Fig.3.7b we observe a high percentage of low goodput experiments.

Fig. 3.7 plots for each experiment the values of $\widehat{G}_C$ on the x-axis, and the values of the RSSI on the y-axis. A first visual inspection indicates that the correlation between the RSSI and $\widehat{G}_C$ values is weak. Using Pearson's correlation coefficient [69] to

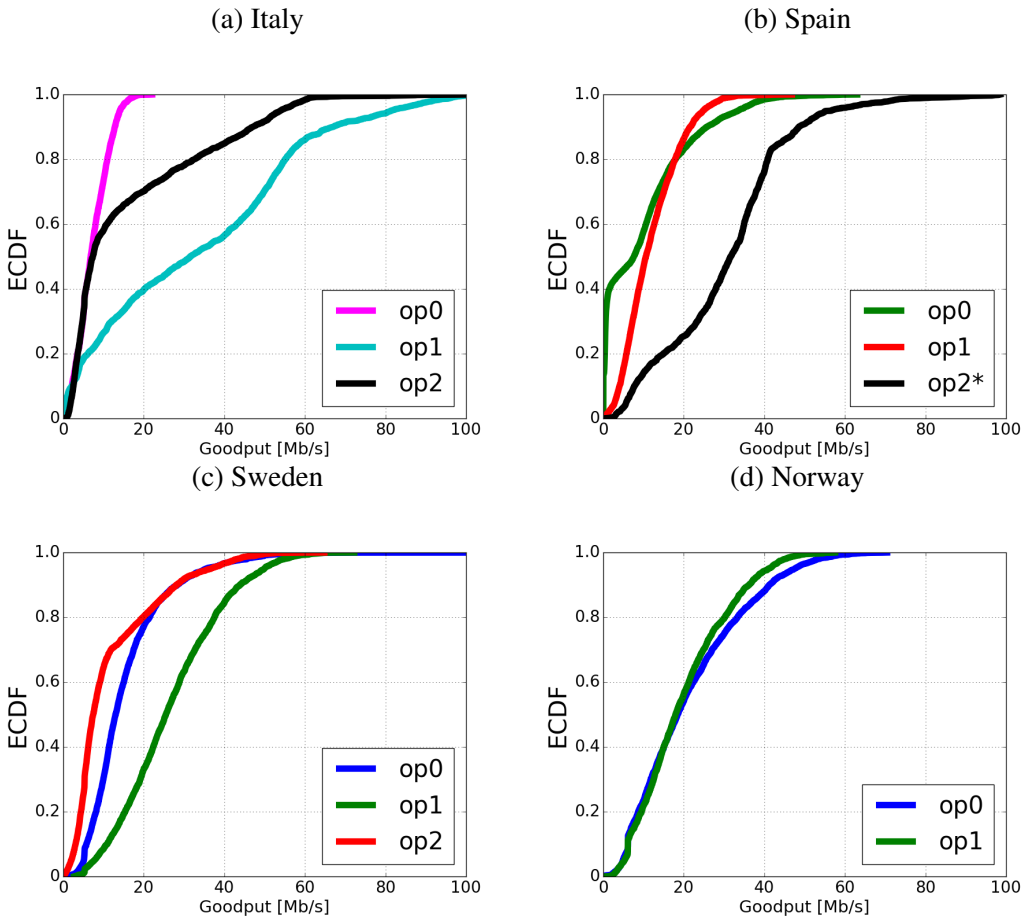


Fig. 3.6 ECDF of the download client-side goodput for the four considered countries

quantitatively corroborate our impression, we obtain values up to 0.37 for Spain and up to 0.61 in Italy (the correlation coefficient takes values in the range [-1,1], with 1,-1, and 0 representing total positive correlation, total negative correlation, and no correlation, respectively). As generally expected, 4G (blue points) frequently outperforms 3G (red points), with some exceptions, which can be explained with the fact that RSSI it is not the only factor determining goodput in a mobile environment.

In Fig. 3.8 we plot for each experiment the average RTT value on the Y-axis, and the RSSI value on the x-axis. Interestingly, from Fig 3.8a, in the case of Italy we can observe two main intervals for RTT values, due to the fact that both op1 and op2 networks are configured so that RTT is mostly less than 50 ms, while op0 provides RTT values in the range of 100 ms. This can be the result of different network configuration choices. In the case of Spain, Fig 3.8b shows that op2*, largely using 4G technology, offers values of RTT in the range of 50 ms, which are lower than

(a) Italy (b) Spain

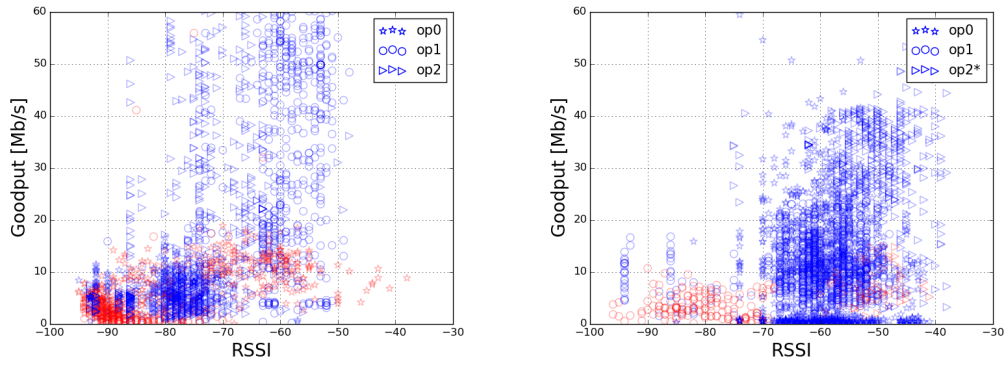


Fig. 3.7 RSSI and download client-side goodput for Italy and Spain. Blue and red markers indicate 4G and 3G, respectively. Pearson's correlation coefficients for Italy op0, op1, and op2 are 0.47, 0.61, and 0.50, respectively. Pearson's correlation coefficients for Spain op0, op1, and op2 are -0.008, 0.37, and -0.02, respectively

(a) Italy (b) Spain

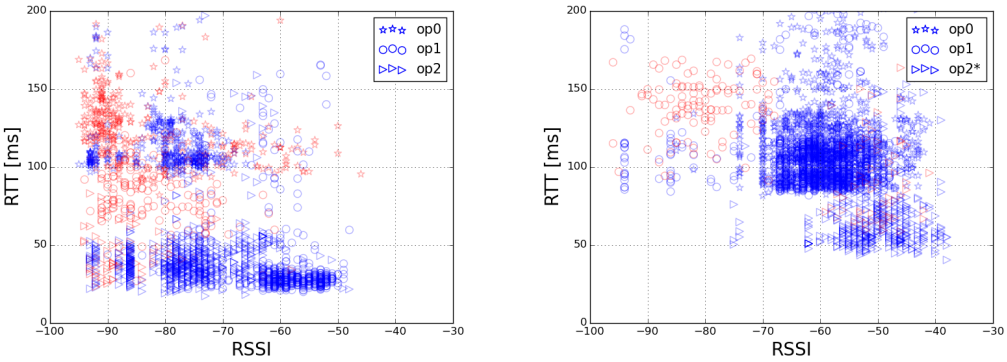


Fig. 3.8 RSSI and RTT for Italy and Spain. Blue and red markers indicate 4G and 3G, respectively. Pearson's correlation coefficients for Italy op0, op1, and op2 are 0.03, -0.49, and 0.39, respectively. Pearson's correlation coefficients for Spain op0, op1, and op2 are -0.009, -0.33, and -0.03, respectively

with other operators. Surprisingly, op2* in Spain is a roaming operator, that offers better performance with respect to the local operators.

3.6.2 Middle Box Detection

Fig. 3.9 shows the goodput in Mb/s experienced from the client-side (x-axis) and the server-side (y-axis), when $IP_C = IP_S$ and $Port_C = Port_S$ for operators in Sweden. If

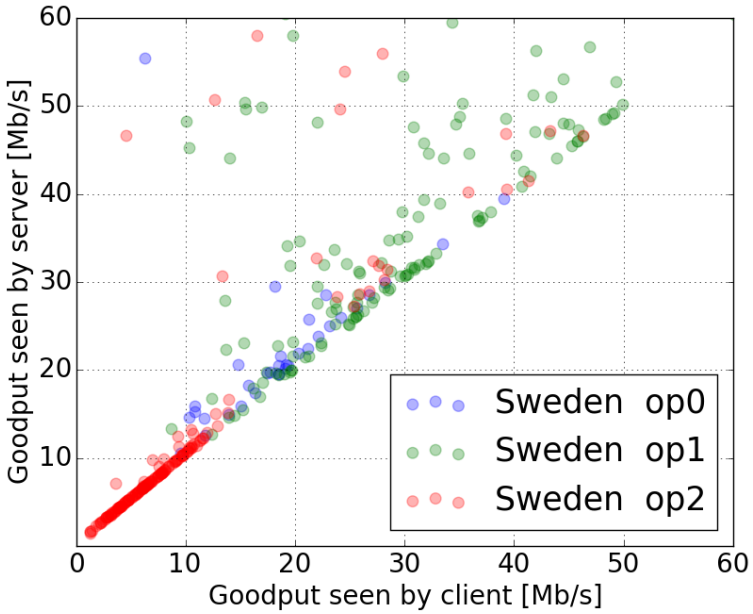


Fig. 3.9 Goodput experienced from client and server sides on Sweden operators

no PEP is present in the operator network, all points are expected to gather along the line $x = y$ in which $\widehat{G}_C \sim \widehat{G}_S$. While we see many points along this line, we also observe points where $\widehat{G}_C < \widehat{G}_S$, indicating the presence of a proxy. This is not surprising, since the use of PEP is becoming a common practice for mobile operators trying to improve end-users' QoE [70, 71, 60].

The MONROE platform allows us to gather detailed information about the operational state of the MBB networks in different countries. For example, we see that the operational setting of the Sweden operators are not static, and change over time. Indeed, the traffic of op2 in Sweden in some time periods crosses a PEP and in some others does not. Fig. 3.10 presents the server-side and client-side goodputs for this operator in the week when the traffic of op2 mostly crosses the PEP. The dashed line (server-side goodput) is often higher that the solid line (client-side goodput), but not always.

The volume of roaming traffic has been steadily increasing in Europe, and will increase even more after the reduction of the roaming surcharges, due to take place in June 2017. Operators have already started offering reduced tariffs for roaming, and exploiting international roaming agreements. In order to look at this aspect of MBB network performance, we considered op2* in Spain, which is the roaming

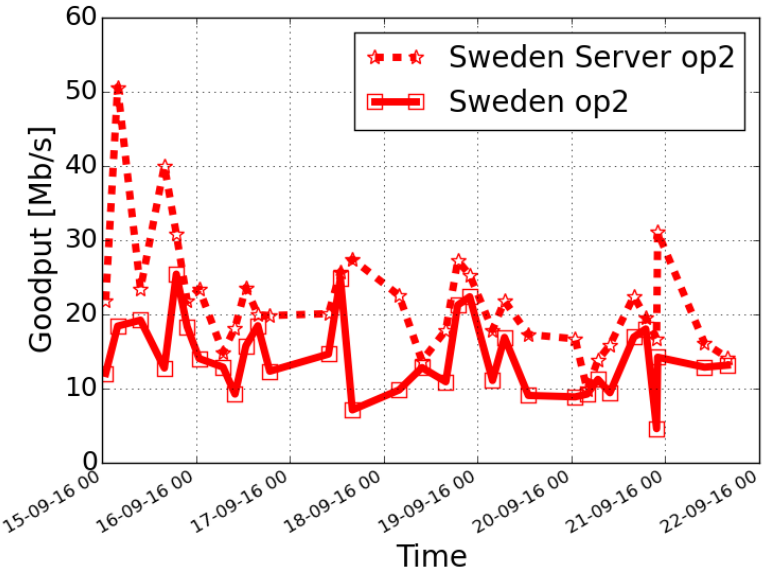


Fig. 3.10 Goodput experienced from client and server sides for op2 in Sweden during one week

network for op2 in Italy. In other words, op2* in Spain is an Italian SIM used in roaming conditions in Spain. Quite surprisingly, Fig. 3.11 shows that the roaming SIM (op2* in Spain) obtains higher goodput than the corresponding SIMs at home (op2 in Italy), and that a PEP is in use in both cases.

Fig. 3.12 shows the values of the MSS and window scaling (WS) declared by the client to the server on port 80. The MONROE platform provides an equal setting at all clients with the default values of 1460 Bytes and 7 for MSS and WS. For visibility, the values in Fig. 3.12 are uniformly distributed around the observed value. Fig. 3.12a shows that Italian operators modify the client-declared TCP options. In order to see this, it is necessary to check more than one option, since, for instance, op1 does not change the MSS value, but changes the WS value. For other operators, the behavior varies. In Spain, both operators keep the WS value, but reduce the MSS value to 1400. In Sweden, operators again keep the WS value, but change the MSS to different values. In Norway, operators always change the MSS value, and sometimes also the WS value.

Finally, Table 3.2 shows a summary of the characteristics observed on the 11 European operators. The third column of the table indicates the usage of the NAT in the operator network. We see for example that in Italy op0 is always using NAT

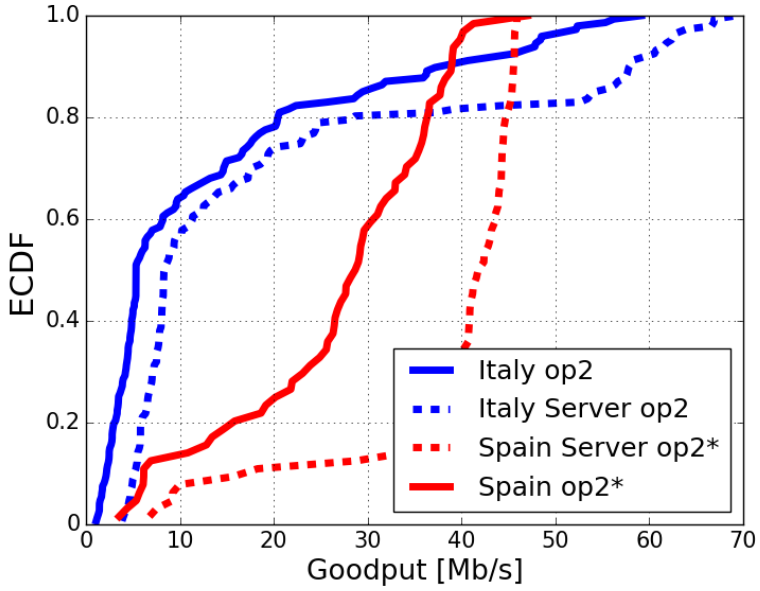


Fig. 3.11 Goodput experienced from the client and the server sides for the same operator SIM in Italy and Spain

(Yes), while op1 sometimes uses it (Yes*), and op2 never uses it (No). Column 4 tells us that most of the operators use a PEP on port 80. The fifth column tells us that all operators do L4 mangling on all tested ports. Column 6 gives the fractions of observed 2G, 3G, and 4G connections.

3.7 Conclusions

In this chapter, we discussed our experience in running ”speedtest-like” measurements to estimate the download speed offered by actual MBB networks. Our experiments were permitted by the availability of the MONROE open platform, with hundreds of multihomed nodes scattered in four different countries, and explicitly designed with the goal of providing hardware and software solutions to run large scale experiments in MBB networks. Our data were collected in 4 countries, over 11 operators, from about 50 nodes for more than 2 months.

Despite their simplicity, download speed measurements in MBB networks are much more complex than in wired networks, because of many factors which clutter

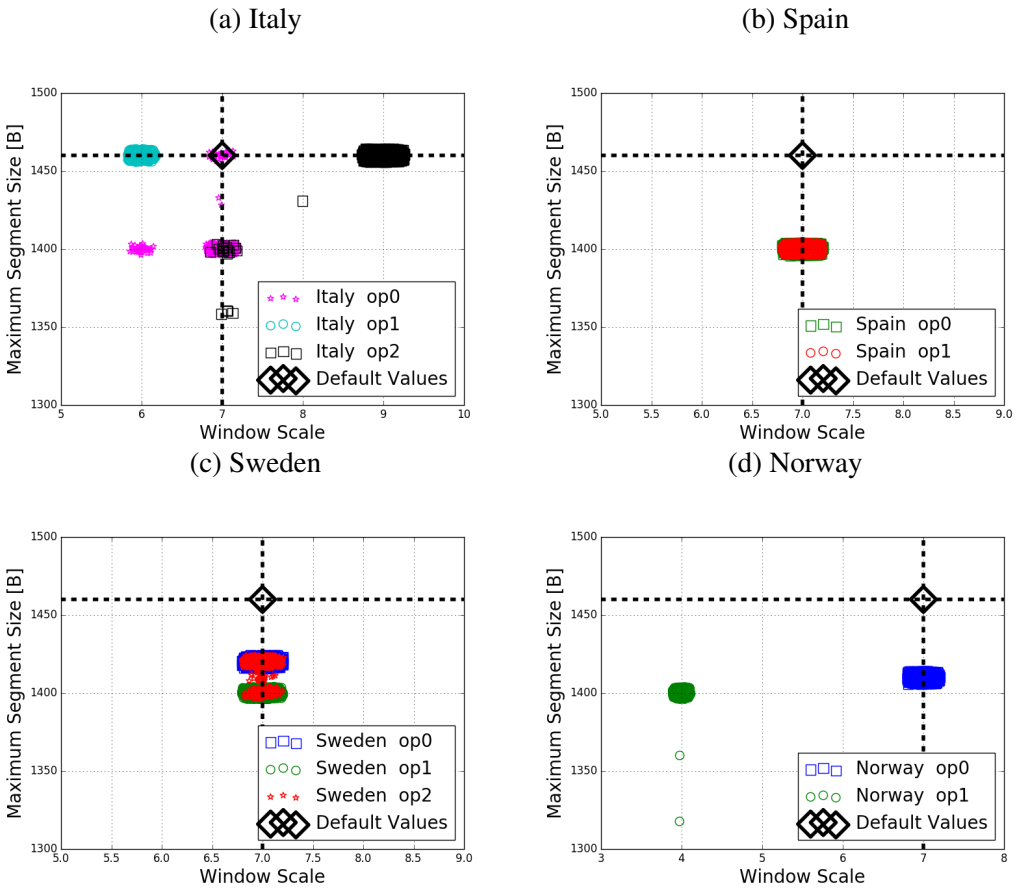


Fig. 3.12 WS and MSS values experienced at the server side on port 80, default values of MONROE nodes are 7 and 1460 Bytes, respectively

the picture. The analysis of the results we obtained indicated how complex it is to draw conclusions, even from an extended and sophisticated measurement campaign.

As a result, the key conclusion of our work is that benchmarks for the performance assessment of MBB networks are badly needed, in order to avoid simplistic, superficial, wrong, or even biased studies, which are difficult to prove false.

Defining benchmarks that can provide reliable results is not easy, and requires preliminary investigation and experience, both being now possible thanks to the availability of an extensive Europe-wide platform like MONROE.

Table 3.2 Summary of the operator settings

Country	Operator	Private IP & NAT	$\widehat{G}$ mismatch on port 80	L4 Mangling	Connection (percentage) Type
Italy	op0	Yes	Yes*	All	3G (0.46), 4G (0.54)
	op1	Yes*	Yes	All	3G (0.15), 4G (0.85)
	op2	No	Yes	All	2G (<0.01), 3G (0.08), 4G (0.92)
Sweden	op0	Yes*	Yes*	All	4G (100)
	op1	No	Yes	All	3G (<0.01), 4G (0.99)
	op2	No	Yes*	All	3G (0.37), 4G (0.63)
Spain	op0	Yes	No	All	4G (100)
	op1	Yes	No	All	3G (0.16), 4G (0.84)
	op2*	No	Yes	All	3G (0.07), 4G (0.93)
Norway	op0	No	Yes*	All	4G (100)
	op1	Yes*	Yes*	All	3G (0.08), 4G (0.92)

Chapter 4

WebWorks: Experimenting the Mobile Web

4.1 Introduction

MBB networks revolutionized the way people interact, bringing a variety of communication services into most of our daily activities. Today, messaging, videos, and the web are key components of our lives, and we expect our MBB network provider to deliver performance and efficiency in highly dynamic scenarios.

With a complex ecosystem of networks, smart devices and traffic-intensive applications, MBB brings both opportunities and challenges to network operators. Indeed, despite many years of mobile networking research and engineering, MBB performance still struggles to meet the growing expectations of users for fast, reliable, and pervasive services. Accurate measurements are necessary to quantify the achieved performance and identify the system bottlenecks. However, the intertwining of technologies, protocols, setups, and service design makes it complicated to design scientifically sound and robust measurement campaigns. For example, the higher the load in the MBB network cell, the larger the variance users perceive in the time to reach the content of interest (e.g., a webpage), which, in turn, translates into poor QoE [2]. In this complex ecosystem, data analytics that focus on finding relationships between user experience and network performance statistics offer the promise of helping operators target those technology improvements that matter most to their customers.

In this thesis, we discuss ways to monitor service performance in MBB networks, with the objective of quantifying end user QoE when user experience is dependent on a large number of factors. We use MONROE system, which we built to enable controlled experiments in multiple MBB networks under similar conditions. Then, by leveraging data analytics, we show how the data we collected enables us to directly relate user experience to network performance statistics, an important step on the way to monitoring and managing service quality and user satisfaction.

In particular, in this chapter we focus on web browsing performance as a study case. Within the MONROE system, we instrumented a long-term, large-scale measurement campaign capable of harvesting cross-layer measurements that capture the complex picture of the current mobile web ecosystem. Today, websites are much more complex than ever, with rich functionality and content. The average number of bytes per page for mobile devices has increased from 872 kB to 2,532 kB in the last three years, which is roughly a 200% increase [72]. In this context, the time a user spends to reach the content (i.e., the web page load time) is a critical metric the carriers aim to optimize [73, 74], because of its strong correlation with user satisfaction and, in turn, with company revenues [75–81].

This chapter demonstrates the potential of MONROE for performance assessment of operational MBB networks. We measure and compare 11 commercial MBB operators in four countries in Europe from the point of view of the web service performance their customers experience. We measure one of the operators also while roaming in another country. We introduce the `WebWorks` experimental setup we built on top of MONROE for mobile web measurements (Section 4.3). We use `WebWorks` to evaluate the performance of using HTTP1.1/TLS (H1s) or HTTP2 (H2) across 10 different target websites by monitoring three different web performance metrics (Page Load Time (PLT), ObjectIndex and ByteIndex). To facilitate this analysis, `WebWorks` logs the browser interactions with the target sites in a JSON-formatted archive file called HAR (HTTP Archive). Based on this, we derive a large number of web metrics, including total web page load time, size of web pages in bytes, number of objects, size of each object, number of domains, object types (javascript, css, image etc.), object load time including DNS resolution, TCP connect, send, wait or object receive timings. We present our unique dataset openly to the community[1]. Our results (Section 4.4) show that, for all the 11 different operators we measure,

[1] https://www.monroe-project.eu/datasets/conext17/

the performance improvement for the H2-enabled websites is very limited. The only website for which the H2-enabled version shows higher performance (according to the PLT, ByteIndex and ObjectIndex metrics) is `youtube.com`. We conjecture this is due to the complexity of the website and the incentives of the corporation hosting it to optimize user experience. Furthermore, we find that the roaming impact is insignificant. Finally, we conclude the chapter in Section 4.5.

4.2 Background and Related Work

In the past years, we have seen increased interest in the networking community from different parties (e.g., researchers, operators, regulators, and policy makers) in measuring the performance of MBB networks. There are mainly three approaches for measuring the performance and reliability of MBB networks: (i) crowd-sourced results from a large number of MBB users [82–84] , (ii) measurements based on network-side data [85, 52–54], and (iii) measurements collected using dedicated infrastructure [86, 55, 56]. In this chapter, we collect data from the MONROE dedicated infrastructure in order to have full control over the measurement nodes, allowing us to systematically collect a rich and a better quality dataset over a long period of time. We focus on web measurements and capturing application performance as experienced by end users for different combinations of target websites and different protocols.

The Hypertext Transfer Protocol, HTTP/1.1 (H1)[87] has been the de-facto standard for loading webpages since 1999. It uses the TCP protocol underneath to deliver web resources to users. However, H1 has been found to limit the performance of web access, specially as todays webpages have become increasingly complex [88]. One key issue with H1 is head-of-line blocking (HoLB) [89], since it practically allows to have only one outstanding request per TCP connection.

The design of web browsers responded to this constraint by introducing parallelism. This workaround allows multiple parallel TCP connections to be used. This alleviates the performance obstacles introduced by the HoLB behavior of H1, but also has its drawbacks. The collateral damage that can occur to other competing TCP flows at an Internet bottleneck influences the available capacity and flow dynamics. Hence web browsers define a limit on the number of parallel TCP connections that a browser can open towards a particular server (six by default in Mozilla Firefox

and Google Chrome). The limitation on TCP parallelism (i.e., the number of TCP connections) has motivated the server to put resources across multiple domains (even for the same server). The web industry have also adopted other techniques like spiriting, inlining and concatenation as best practices, all of which have their own shortcomings [88].

In light of the above observations, Google proposed SPDY [74] to reduce the web latency keeping the original semantics of H1. Subsequently, IETF standardized HTTP/2 (H2) [73], the evolution of HTTP based on SPDY. H2 uses a single multiplexed connection per domain, allowing also request prioritization between multiplexed requests. This reduces overhead and limits HoLB. Efficiency is further optimized through the use of header compression.

Throughout the recent years, several studies have been tracking and reporting the adoption of H2 in the web [90–92]. The relative performance differences between H1 and H2 have been receiving much attention, with numerous related research efforts in this direction [91, 93–101]. The results are mixed, hence it is hard to make a clear conclusion on which protocol outperforms the other and in which scenarios. Research reported in [95, 96, 100, 98, 94] primarily used lab setups with emulated network scenarios. [97, 100] considered one single operator network to measure the web performance using both dummy pages and real archived pages or real sites. [102] models H2 performance using H1 traces, providing an upper bound of H2 performance. [90] identified the lack of key H2 features in its current adoption as a reason for the absence of performance improvements in H2 measurements. In this chapter, we present a large-scale web measurements campaign using MONROE, including results for 11 different mobile carriers over four countries. We do not only measure the protocol difference from different vantage points, but also set the operators network performance in a common scale.

Previous web performance measurements also differ in the metric they chose to evaluate performance. Work in [95, 96, 100, 92, 97, 90] used PLT, a metric primarily based on OnLoad event fired by the browser. This event is fired when all objects on the page are loaded. PLT and similar metrics have been criticized for not being representative of user experience [103–106]. Users are mostly interested in above the fold content (AFT), PLT waiting for the OnLoad event when all objects are loaded thus overestimates the user perceived latency [105, 106].

Google introduced SpeedIndex [107, 108], as an alternative to PLT to better capture the user perceived experience. SpeedIndex is a measure of an average time to get all AFT in the screen, in other words an average time for the visual completion of a page in the browser. More visual contents at the beginning of the page loading process lead to smaller SpeedIndex. However, measuring SpeedIndex requires to film the page loading process and is thus quite complex and can significantly inflate the measurement time. Similar but much less computationally intensive metrics, ObjectIndex and ByteIndex are proposed in [105]. Additionally, [109] proposed the "3rd Party Trailing Ratio" metric, which measures the fraction of download time for the 3rd party assets on the webpage critical path. The authors show that this is another determinant factor influencing web performance. In our measurements, we consider three different metrics, namely ByteIndex (BI), ObjectIndex (OI) and PLT.

More then simple measurement analysis, we tackle the need for understanding the relationship between web performance and network characteristics, which is important for cellular operators looking to tackle network conditions that impact web. For example, while generating a model to determine a web-page's QoE, A. Balachandran et al. [110] found evidence of the impact of the number of users and radio access technology (RAT) handovers on the web-page performance. They discovered that the performance is greatly affected by a website's own complexity (e.g., number of objects, domains visited) with some influence of the time-of-day feature. Their study concluded that when web QoE has logical values like yes/no (e.g., abandonment of session (yes/no), partial download exists in session time (yes/no)) then decision tree algorithms work well. However, when the QoE is measurable (e.g., partial download ratio (i.e., ratio of objects that do not complete download in a session), or session length (time when user is on a web-page)), linear regression modeling gives the best result. Our results confirm these previous findings and bring to notice the complex correlations between various aspects of the web browsing process and the experience of the end users.

4.3 Measurements

In this section, we describe the approach we put forward for measuring web performance in MONROE. We built our web performance measurement platform on top of the MONROE system. We design and implement `WebWorks`, the MONROE-

compatible Docker container that measures web performance while visiting popular webpages using Firefox in headless mode. Below, we describe the methodology we integrate in WebWorks, the specific experimental setup we employ for running our measurements and the dataset we collect.

4.3.1 WebWorks Design

In this section, we describe the internals of the WebWorks experiment methodology, as well as the metrics we collect. WebWorks[2] enables the collection of multiple web performance metrics while visiting a target webpage using Firefox in headless mode. We leverage the Selenium web automation framework [111] to simulate web surfing and collect web performance metrics in MONROE. Among the several tools that the framework provides, the Selenium webdriver offers a large set of APIs to interact with a given web browser in the same way as a regular user would. For example, we use the APIs to click on links, buttons or to enter text in input forms. Selenium is compatible with a number of browsers (e.g., Internet Explorer, Firefox, or Chrome) and also provides a number of language bindings (e.g., C#, Java, JavaScript, or Python). We enable WebWorks to use Selenium with Firefox: upon invoking the webdriver, WebWorks launches the native Firefox browser in the MONROE nodes to visit any target input webpage. We set the user agent string in Firefox as to retrieve mobile versions of the pages from the web servers. Additionally, to capture the user web experience as realistically as possible, when creating the Firefox profile during a web page visit, we also provide cookies and login information (if applicable) to Selenium. MONROE nodes are not equipped with displays that GUI-based programs like Firefox require to render the output. We thus use the X virtual framebuffer (Xvfb [112]) to mimic the missing display and enable the browser to behave normally.

During each experiment run we use the HAR export trigger add-on [113] to log Firefox's interactions with the visited pages in a JSON-formatted archive file called HAR (HTTP Archive). We then use the HAR file to derive a number of web performance metrics, including PLT, size of web pages in bytes, number of objects, size of each objects, and number of domains. Additional metrics such as object types (javascript, css, image etc.), object load time including DNS resolution time, TCP

[2] WebWorks is available in https://github.com/MONROE-PROJECT/Experiments/

connection time, and object receive timings are also available from the HAR files. Apart from the HAR, we collect RTT statistics during each experiment run.

`WebWorks` tracks three different metrics, namely BI, OI, and PLT. The main performance metric is PLT, a metric primarily based on OnLoad event triggered by the browser. This event is fired when all objects on a page are loaded. Furthermore, we infer the OI and BI [105] from the HAR files. They are computed from the arrival time of all objects in the webpage waterfall. OI tracks the time at which the content of the page is retrieved, taking into account all external images, style-sheets and scripts needed to render the page. BI operates in the same way, but weights objects by their size. A higher value indicates higher page load time.

4.3.2 Experimental Setup

In this section, we present the experimental setup of the measurement campaign we run with `WebWorks` on MONROE. We deploy the `WebWorks` Docker container on 18 MONROE nodes (each node measuring up to three different mobile carriers at the same time) that operate in the four countries with MONROE coverage (i.e., Norway, Sweden, Italy, and Spain). For the measurement campaign we analyze in this chapter, we selected specific nodes which operate inside university campus areas, ensuring consistency in terms of end-user traffic patterns across countries.

When running the measurements on a MONROE node, `WebWorks` first verifies which are the existing MBB connections on each node (i.e., the MBB operators we can experiment with on each node). It does so by listening to the active stream of metadata continuously transmitted within the MONROE software ecosystem, as previously explained in Section 2.4. Each node can measure up to three different mobile carriers. For each active mobile connection detected on the node, `WebWorks` sets as the source endpoint the interface that corresponds to the current operator it selected. It then starts measuring all target websites in an input list we provide, following a random order. MONROE nodes resolve the target websites using Google's public DNS resolver; not the mobile carrier's default resolver (during this measurement campaign).

For this measurement campaign, we configure `WebWorks` to collect web performance measurements while visiting 10 popular websites. We list the websites of interest in Table 4.1. We selected targets that provide H2 access and that are listed

Table 4.1 Characteristics of the target websites we select for our measurement campaign. These are average values over the 10 different pages we visit per website

Site	Size (KB)	# Objects	# Domains
facebook	798	76	6
instagram	1,230	33	6
youtube	815	30	9
wikipedia	241	10	3
google	114	13	4
linkedin	232	24	5
yahoo	1,480	49	8
ebay	493	28	11
guardian	1,895	133	33
nytimes	3,131	205	55

among the most viewed sites in the Alexa [114] top ranking. As all the target websites we chose also expect TLS connections by default, we enable WebWorks to run over H1s, instead of simple HTTP1.1. We also ensure that with our selection of measurement targets we cover a wide range of user interests in terms of topics, including social networking, video, career, search engine, news site, wiki, or shopping. For each target website, we chose 10 different pages to visit (i.e., 100 pages in total) in order to capture a wide range of resource sizes, resource counts and domains visited. For example, instead of measuring the landing page for facebook.com, we visit specific target pages, such as 'facebook.com/telia/' or 'facebook.com/LeoMessi/'. We present statistics per target website in Table 4.1.

For each target website, WebWorks triggers Firefox to download the associated pages we select, also in a random order. We enable Firefox to cache the each page visit throughout the time it measures the same mobile operator. When moving on to a different operator connected on the same node, Firefox clears the caches before starting the measurements against the full set of targets.

4.3.3 WebWorks Dataset

We ran our measurement campaign in May 2017 (from 30th of April until 17th of May) and June 2017 (from 1st of June until 14th of June). In total, we monitor 11 mobile operators, which we list in Table 4.2. We collected more than 50,000

Table 4.2 Statistics on the `WebWorks` dataset; the Country shows where the subscription is active

Operator Name	Country	# measurements
Telia (SE)	Sweden	6,473
Telenor (SE)	Sweden	6,345
3 (SE)	Sweden	4,549
Telenor (NO)	Norway	6,350
Telia (NO)	Norway	2,806
ICE (NO)	Norway	2,664
TIM (IT)	Italy	4,392
Vodafone (IT)	Italy	1,961
Wind (IT)	Italy	1,883
Yoigo (ES)	Spain	3,183
Orange (ES)	Spain	727
Telia (SE)	Norway	7,901
Vodafone (IT)	Spain	1,093

samples from the 18 MONROE nodes we instrumented for this work, each node with at most three different MBB interfaces active. We show the distribution of samples per operator in Table 4.2. This proves the capability of the MONROE system to enable repetitive measurements within the same context, something that is challenging for crowd-sourcing approaches to achieve. The `WebWorks` dataset is balanced in terms of the number of measurements for H1s/H2 protocols. Aside from measuring native operators within their home countries, we also measure roaming. Specifically, we run `WebWorks` using a Telia (SE) mobile subscription roaming in Norway and a Vodafone (IT) subscription roaming in Spain. Our dataset includes approximately 8,000 samples where the SIM card was in roaming mode.

Dataset limitations: We collected the dataset using a limited number of MONROE nodes (18 out of 150 available). The nodes we selected are operating under similar conditions. Usually, campus areas are locations with good coverage that present similar diurnal traffic patterns throughout the countries we measure. This allows us to compare the performance of operators within these constraints. We expect performance to vary as we also vary the geographical distribution of the vantage points. We leave for future work the analysis of a measurement campaign with measurements from a wider variety of spatio-temporal settings.

Whenever measuring a MBB operator, we use commercial-grade mobile subscriptions that are compatible with the ones customers can purchase. The differences in commercial offers thus reflect in the different data plans we activate in MONROE and in the dataset we collect. For example, we observe in Table 4.2 that the number of samples for Orange (ES) (limited data plan, 10GB) is much smaller than the number of samples we collected for Telia (SE) (unlimited data plan, 200GB). For this reason, we are not able to run the modeling analysis (Section4.4.2) for all operators, and eliminate Orange because of the lack of large number of samples.

Proxy detection: To get further insight about the existence of a proxy in the MBB network, we also visit all pages for each of the 10 target websites from each of the MONROE nodes using a wired connection. For each TCP connection, we collect information about TCP options such as Timestamps, Maximum Segment Size (MSS), and negotiated window scale factor. Since we have a wired connection, we can determine which options are declared and supported by the web servers. If the ISP does L4 mangling, or a PEP is present on the path, this can be detected by comparing different TCP options on the wired and wireless connections for a given web page, also when the requests are served by the same IP. Our analysis show that all the 11 operators (see Table 4.2) in our dataset do L4 mangling.

4.4 WebWorks Results

In this section, we provide an overview of web performance in MBB networks in Europe. We then present a more detailed analysis of the relative performance between H1s and H2.

4.4.1 Web Performance Analysis

We first take a look at the web performance as a whole, without differentiating the websites. We illustrate in Fig. 4.1 the country-wise overall operator performance. Here, we consider all the three metrics for the performance evaluation: PLT, BI, and OI. We observe that the different metrics provide consistent results and the overall web performance is similar across different countries and operators, with only slight variations. At this aggregate level, we also observe similar performance between

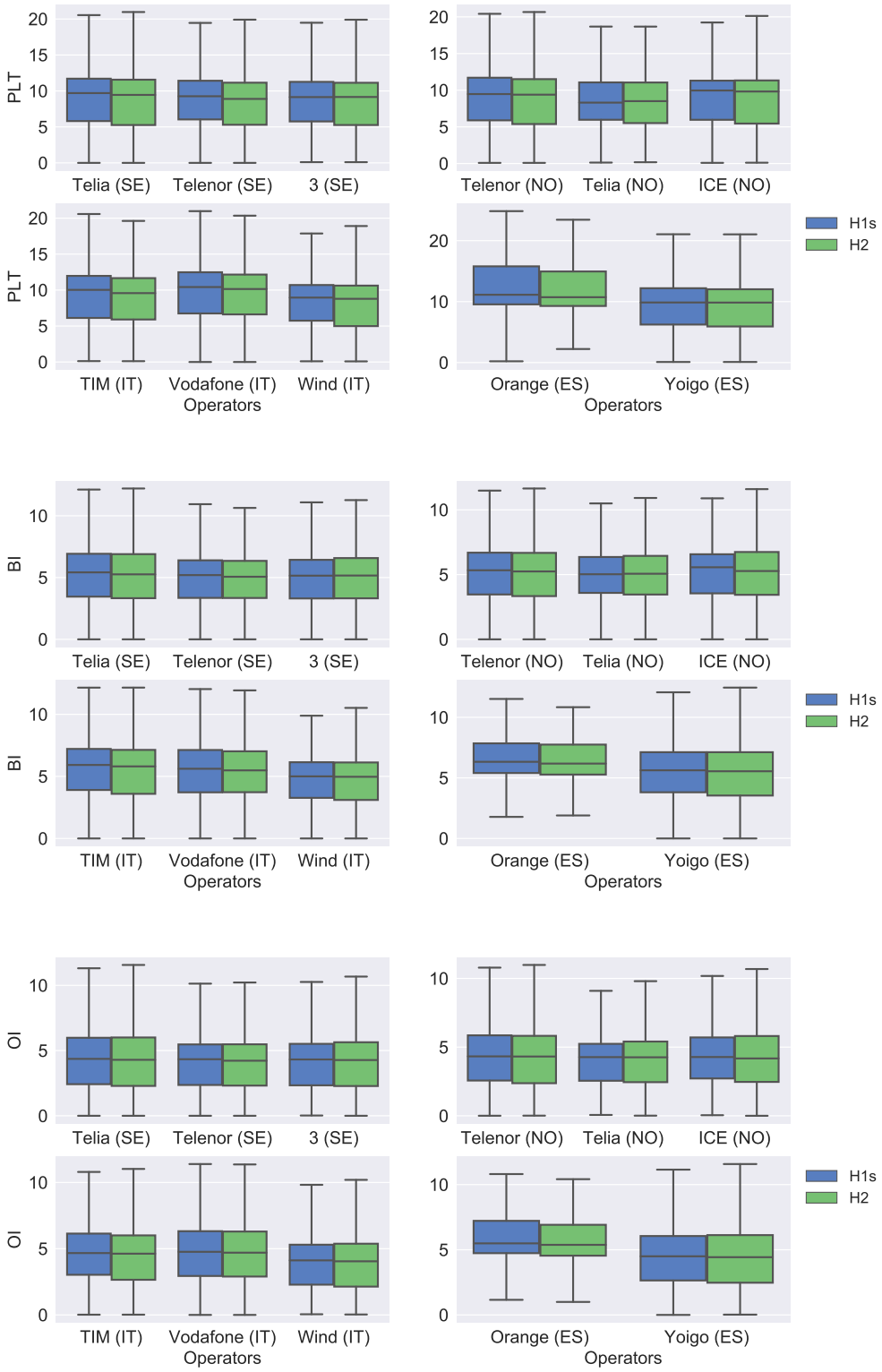


Fig. 4.1 Country wise overall operator's performance considering PLT, BI, and OI

H1s and H2, indicating that the choice of protocol has no significant impact on performance.

In Fig. 4.2, we go into more detail, and report the performance of individual websites on each MBB network. We selected 6 websites among our set of 10, and we only show the PLT metric, due to space constraints. As expected, we can see from the Fig. 4.2 that the PLT varies greatly between the different websites. For instance, we measured median PLTs for wikipedia of around 7 s, and median PLTs for nytimes of around 30 s or above. Contrariwise, we see very limited differences between operators. However, we can see from the box plots that the measured performance for a given operator and website varies significantly between the experiments, indicating the complexity and dynamics of MBB networks. Looking at the relative performance of H1s and H2, we see that there appears to be no consistent difference between the two protocols also when examining the results on a per website granularity with the only exception of youtube.

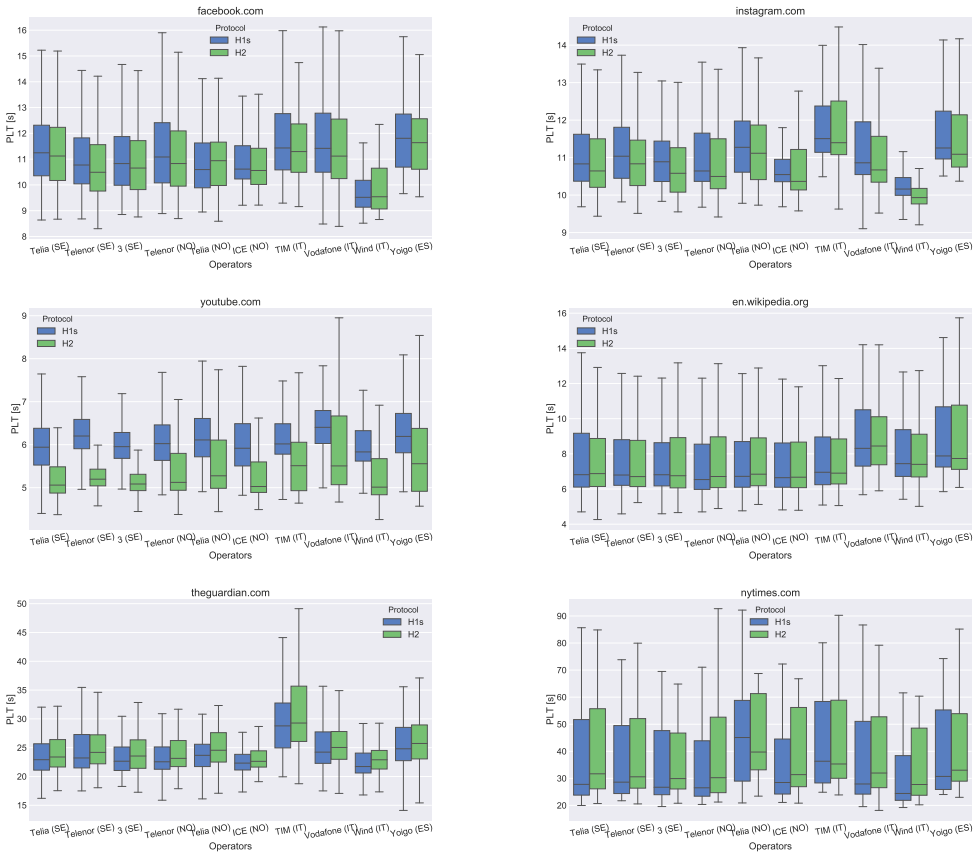


Fig. 4.2 Per website PLTs for different operators and websites

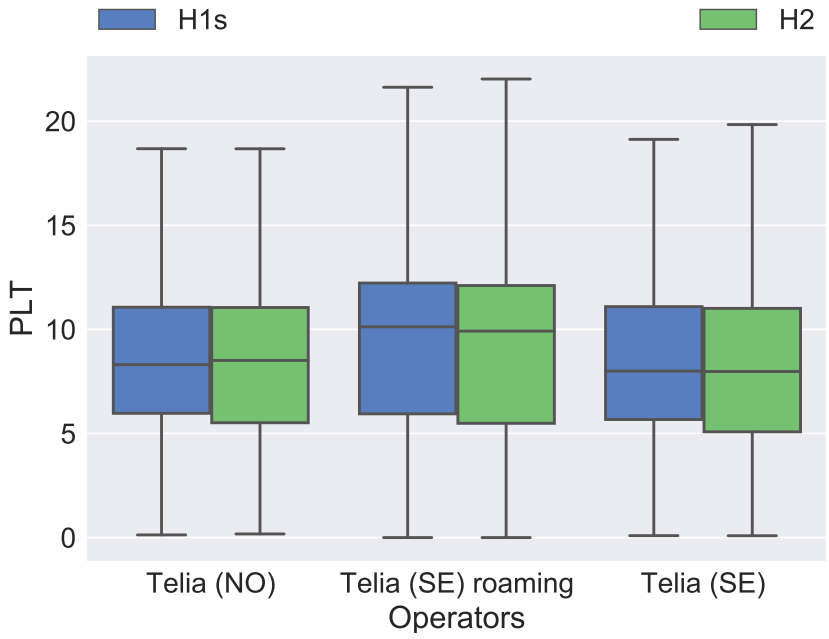


Fig. 4.3 Influence of international roaming on performance. We compare two native operators (Telia (SE) and Telia (NO)) with the case of Telia (SE) roaming in Norway (the visited network is Telia (NO))

Finally, we consider the roaming scenario. To this end, we used a Swedish Telia subscription roaming in Norway and compare it with the performance of Telia when used as a native operator in Norway (Telia (NO)) and Sweden (Telia (SE)). Our preliminary analysis shows that for this roaming scenario, all roaming traffic is detoured over Sweden (home routing). To observe whether this detour in routing has any impact on the web performance, we compare the three subscriptions in Figure 4.3. We observe that the roaming scenario is very similar to the native operator scenarios. Considering that the distance from Norway to Sweden is small, the impact of roaming is negligible and the users can enjoy a similar web experience while roaming. Further analysis is required to understand whether the roaming has a larger impact in the case of countries that are farther apart.

Note that our goal in this section is to compare operators' performance within the constraints of our dataset, which only considers measurements collected from a small geographic area. Thus, our aim is to highlight the complexity and dynamic nature of MBB networks, even when measured within a controlled environment such as MBB.

4.4.2 H1s vs H2 Performance

An overview of our measurement results was provided in the previous subsection. Here we delve deeper into the performance of H1s and H2. In order to quantify the performance difference between the two protocols, we apply statistical analysis techniques to compare measurements referring to different datasets.

We first estimate the Empirical Probability Density Function (EPDF) and ECDF for the H1s and H2 datasets. Using a bin size of 100 milliseconds with support range of $[mean - 3 * std, mean + 3 * std]$ for each website, we compute the frequency of samples falling in each bin. In the literature, there are different well-known Statistical Distance Metrics (SDM), each with its own properties and limitations. In this work, we chose the Jensen-Shannon divergence (JS_{div}), which is defined as:

$$JS_{div} = \sum_i \Big\{ \frac{1}{2} p_i \ln\Big(\frac{p_i}{\frac{1}{2}p_i + \frac{1}{2}q_i}\Big) + \frac{1}{2} q_i \ln\Big(\frac{q_i}{\frac{1}{2}q_i + \frac{1}{2}p_i}\Big) \Big\}$$

Where p_i and q_i (relating respectively to H1s and H2) are the EPDF values generated by samples falling in the i-th bin. JS_{div} is a statistical measure based on the Kullback-Leibler divergence. JS_{div} adds symmetry (i.e., $JS_{div}(p,q) = JS_{div}(q,p)$), and bounded image (i.e., $JS_{div} \in [0, \ln(2)]$), to the Kullback-Leibler divergence. In fact, the reason we chose the JS_{div} is precisely to obtain a symmetric bounded value for our comparisons. JS_{div} is equal to 0 if $p = q$, while it reaches $\ln(2)$ for two completely disjoint distributions.

In Appendix A, we deeply investigated the criteria for choosing the SDM and the threshold for JS_{div}. We selected the threshold values $T^- = 2/100$ and $T^+ = 1/10$. Intuitively, when $JS_{div} \in [T^+, \ln(2)]$, the difference between the two EPDFs (populations) is significant. When $JS_{div} \in [T^-, T^+)$ the difference is observable, and negligible if $JS_{div} \in [0, T^-)$.

Table 4.3 reports the values of JS_{div} for each operator, independent from the website. The JS_{div} is computed over the two EPDFs of H1s and H2 for the three performance metrics BI, OI, and PLT. Since the value of JS_{div} is sensitive to the number of samples, operators/websites with not enough samples are not presented in Tables 4.3 and 4.4 [3]. Table 4.3 shows that none of the considered operators exhibits

[3] We consider the variable support value for each website in range of $[mean - 3 * std, mean + 3 * std]$ with bins of 100 millisecond, Therefore, we remove populations with less 1000 samples.

Table 4.3 Jensen-Shannon divergence of BI, OI and PLT comparing H1s vs. H2 for each operator

Operator	BI	OI	PLT
TIM (IT)	0.058	0.061	0.061
Yoigo (ES)	0.06	0.053	0.091
3 (SE)	0.046	0.036	0.066
Telenor (SE)	0.036	0.029	0.061
Telia (SE)	0.041	0.037	0.062
ICE (NO)	0.077	0.066	0.097
Telenor (NO)	0.04	0.033	0.056
Telia (NO)	0.048	0.045	0.066

Table 4.4 Jensen-Shannon divergence of BI, OI and PLT comparing H1s vs. H2 for each website

Operator	BI	OI	PLT
ebay.com	0.023	0.019	0.034
en.wikipedia.org	0.009	0.01	0.01
facebook.com	0.007	0.007	0.013
instagram.com	0.064	0.048	0.029
linkedin.com	0.066	0.027	0.031
theguardian.com	0.047	0.047	0.059
youtube.com	0.033	0.063	0.207

a significant difference between H1s and H2. This is consistent with the overview of the measurement results shown in Fig. 4.1.

Table 4.4 reports the values of JS_{div} for each website, independent of the operators. Youtube exhibits a value $JS_{div} = 0.207$ for the PLT metric, hence falling in the range of significant difference. In Addition, Fig. 4.4 shows that in this case H2 yields lower PLT with respect to H1. While we observed some differences for Youtube, the analysis of other cases confirms that no consistent difference can be observed between H1s and H2 in our measurements. One reason that has been identified in a very recent work could be that some of the key H2 features are still not in use at large in the server side [90]. Also we have seen in our findings that domain wise resource placement is quite similar for both the H1 and H2 cases in the server side, which influences the potential gain from multiplexing in H2.

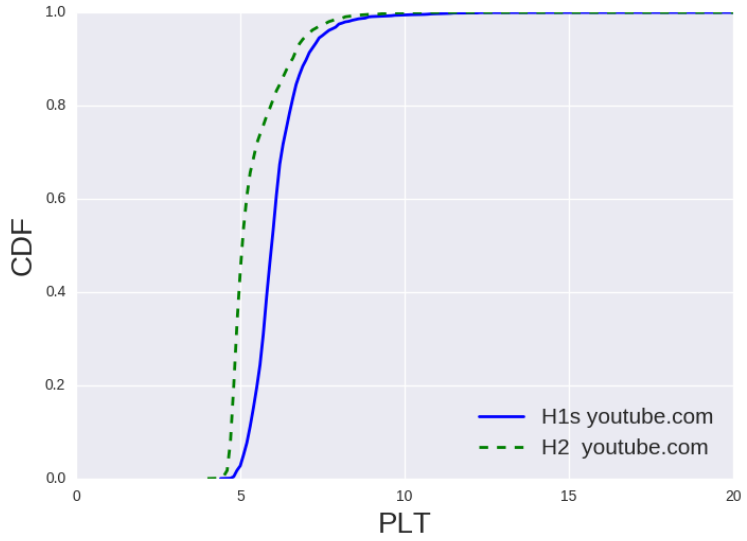


Fig. 4.4 The ECDF of the PLT of Youtube for H1s/H2

4.5 Conclusions

This chapter presented a cross-European study of web performance on commercial mobile carriers using the MONROE system. The novelty of the study stands in the sheer volume of data we were able to collect from MONROE nodes operating under similar conditions in 11 different MBB networks. Our results and further analysis brought to light the complexity of the cellular networks, where the randomness of the wireless access channel coupled with the often unknown operator configurations makes monitoring performance very challenging. We find that the overall web performance is similar across different countries and operators, with only slight variations. In aggregate per target website, our measurements show that the performance improvements H2 promised still remain to be experienced. Furthermore, we find that web performance is mainly dependent on the characteristics and performance of the target web page. Thus, for websites where we conjecture that the server-side implementation of H2 is more mature (youtube) we observe superior performance from the end-user perspective.

Chapter 5

Understanding Roaming in Europe

5.1 Introduction

International roaming allows mobile users to use their voice and data services when they are abroad. The EC, in an effort to create a single digital market across the EU, has recently (as of June 2017) introduced a set of regulatory decisions [12] as part of the "Roam like Home" initiative. This initiative abolishes charges for users when they use voice and data services while roaming in EU. In this setting, Mobile Network Operators (MNOs) are expected to deliver services with QoS properties similar to the ones a user experiences when at home.

To support roaming, MNOs commonly connect with each other through an IP Packet Exchange (IPX) network. An IPX [115, 116] can be described as a hub that interconnects MNOs over a private IP backbone network and is possibly run by a third party IPX provider. An IPX provider has connections to multiple network operators and thus enables each MNO to connect to other operators via a single point of contact. Fig. 5.1 presents a set of topology architectures that can be used for roaming in a mobile network, namely, HR, local breakout (LBO) and IPX hub breakout (IHBO).

When a mobile node is at home (left, see Fig. 5.1), the *home user*'s traffic will take a short path inside the network to reach a suitable Packet Data Network Gateway (PGW) to the Internet. The traffic of a *roaming user* (right, see Fig. 5.1) is directed to an egress PGW whose location depends on the roaming architecture. In the case of HR, the mobile node receives the IP address from its home MNO and the

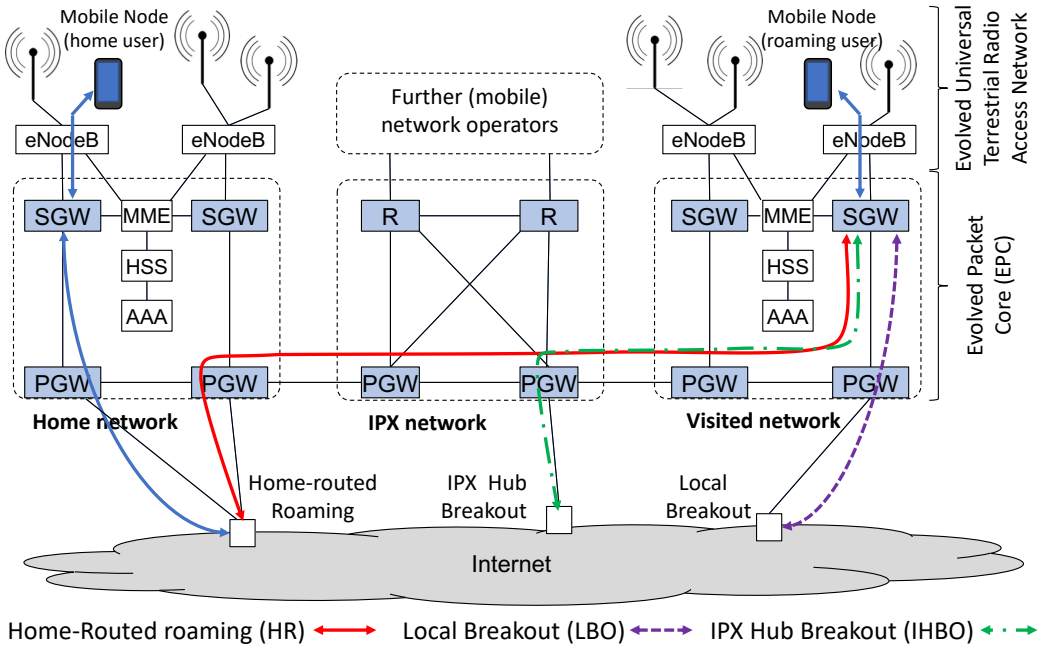


Fig. 5.1 Internet access options for a mobile node at home (left) and when roaming (right)

roaming user's traffic is first routed towards a PGW in the *home network* (red path). With LBO, the mobile node receives its IP address from the visited network and the traffic is routed towards a local PGW in the *visited network* (purple path). When using IHBO, the mobile node obtains its IP address from the IPX networks and the traffic is routed through a PGW in the IPX network (green path). These approaches have a potential impact on the communication performance. For instance, when the node accesses services inside the *visited network*, the performance is likely to be worse in the HR case, because all packets travel twice between the visited and the home country; less so when the communication peer is in a third country and minimally when accessing services in the home country.

In this thesis, we perform an extensive large-scale measurement study[1] to understand the roaming ecosystem in Europe after the "Roam like Home" initiative. More specifically, we investigate: (i) Which technical solutions are actually being deployed and used today? and (ii) What are the implications of roaming on the service experienced by the roaming user?

[1] We will make the code and the dataset we collected open to the community upon publication.

To address these questions, we built a unique measurement platform, *EURoam*, to assess roaming and its performance implications. The platform integrates dedicated measurement hardware that we deployed in six different countries (see Fig. 5.3 and section 5.2 for details) across Europe, covering a total of 16 MNOs. We purchase Subscriber Identity Modules (SIMs) that support roaming for these MNOs and distribute them across the six countries. We characterize roaming operation and network performance (section 5.3) and evaluate the impact on VoIP and web applications (section 5.4) while roaming. We find that all observed MNOs use HR, which yields noticeable latency increases. We do not observe traffic differentiation policies for VoIP or web, but we do find evidence of content discrimination for roaming users. We review the existing work in section 5.5 and conclude the chapter in section 5.6.

5.2 EURoam and Measurement Setup

In this section, we present the hardware platform we built for roaming measurements, and the manner in which we orchestrate it to run measurements and collect our data.

5.2.1 EURoam Platform

We design and build EURoam, a dedicated platform for roaming measurements in Europe. EURoam integrates several components that we depict in Fig. 5.2. The main blocks include measurement nodes distributed in six different EU countries, the backend system, several measurement servers and a scheduler, all of which we detail next. To build the EURoam platform we adapt the open source software provided by MONROE [117], an open measurement platform.

EURoam nodes: Each EURoam node is composed of PC equipped with two 3G/4G MC7455 LTE CAT6 miniPCI express modems. Because of the high cost of nodes and subscriptions, and the complexity of the coordination effort required (see subsection 5.2.3), we have setup a small scale platform with a total of 12 EURoam nodes dedicated for roaming measurements.

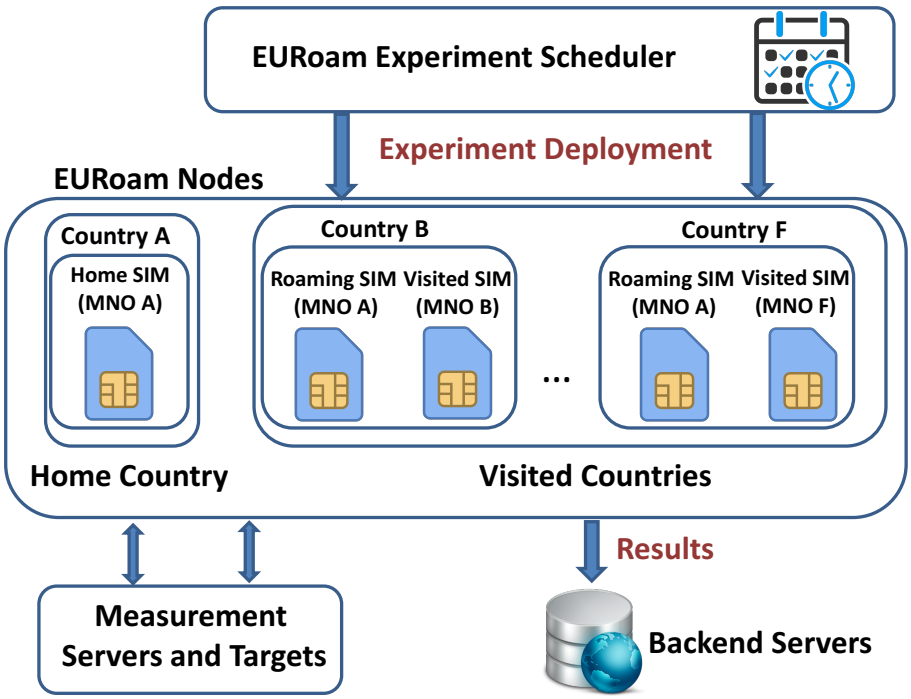


Fig. 5.2 EURoam platform and experimental setup

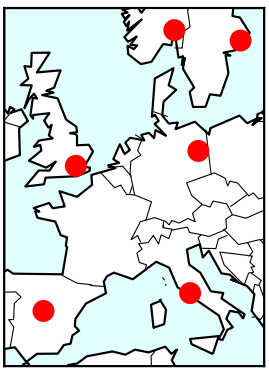
EURoam backend: Upon completion of each measurement, EURoam nodes transfer the results of the measurements to a central server for further processing analysis.

Measurement servers: We have deployed one measurement server in each country as measurement responders and also to capture traffic traces.

EURoam scheduler: The scheduler allows the user to query for resources, select nodes and launch different tests in the platform simultaneously. We used the open source MONROE scheduler as a basis for the EURoam scheduler. Each test is designed and implemented in a Docker container [29].

5.2.2 Experimental Setup

We deployed two EURoam nodes in each of the six European countries and we measured a total of 16 MNOs that operate their own network, as illustrated in



		MNOs	
NO	Telia NO	Telenor NO	
SE	Telia SE	Telenor SE	Tre
UK	Vodafone UK	EE	
DE	Vodafone DE	T-Mobile DE	O2 DE
ES	Vodafone ES	Movistar	Orange
IT	Vodafone IT	TIM 3	3 IT

Fig. 5.3 The distribution (left) of the EURoam nodes in six countries and (right) SIMs for 16 MNOs we measure across Europe. Each country deploys two EURoam nodes and one measurement server

Fig. 5.3.[2] Note that in UK and Norway, there are two major MNOs, while in the other countries we have considered three MNOs.

For each MNO, we bought six SIMs that support roaming in Europe and we distributed one SIM in each of the countries we cover. For example, for a given MNO A in country A, we distributed five *roaming SIMs* of MNO A to countries B-F and kept another *home SIM* in the home country (e.g., country A). Each roaming SIM connects (or *camps*) to a local roaming partner (or visited network) native to the visited country.[3] For each roaming SIM, we identify the corresponding visited network (e.g., MNO B in country B) and, when available, activate the corresponding SIM from the visited network (which we hereinafter denote by *visited SIM*). Fig. 5.2 illustrates the configuration in our experimental setup.

5.2.3 Measurement Coordination

Each MNO-specific measurement round involves 6 nodes: (i) one node with the home SIM and (ii) five nodes with both the roaming SIM and the corresponding visited SIM. This enables us to capture performance metrics for the roaming SIM, but also to compare those with the local performance of the home network and the visited network (when possible).

[2] In the figure, we illustrate the capital of each country as the corresponding node and measurement server location in order to anonymize the exact locations of our measurements.

[3] When MNO A (from country A) is a roaming partner of MNO B (from country B), MNO B can serve MNO A's customers roaming in country B by allowing MNO A users to camp on MNO B's network (and vice-versa).

Before running the set of measurements (see section 5.3 and section 5.4), we first need to configure the nodes by activating and deploying the SIMs. For each MNO, we carry out the measurements at the same time from all six countries and we coordinate the configuration of the experimental setup in two steps:

Home and Roaming User Activation: Given an MNO, we first insert the SIM into the first SIM slot in each node in all six deployment locations. For the SIM located in its home country, this step triggers the home user activation (by inserting the SIM in the measurement node), while for the rest of the nodes, the roaming user activation is triggered.

Visited User Activation: Once we complete the home and roaming user activation, we check the visited network that the roaming SIM uses in each of the five visited countries. Then, we insert the SIM of this MNO (when available) into the second slot of each corresponding node.

Using the EURoam scheduler, we orchestrate the execution of the measurements so that they run in parallel on all nodes. The measurement coordination effort was a significant part of the process. In each country, at least one person was dedicated to carry out the experimental setup configuration for each MNO in a timely manner. Given that we deploy two nodes per country, we could measure two MNOs in parallel. We coordinated the SIM changes over email. Furthermore, before the change of the next pair of SIMs, we double-checked the measurement results we had collected to ensure correctness and completeness of the dataset. Each round lasted one week, over a total period of more than 16 weeks of experiments.

5.3 Roaming Setup and Performance

5.3.1 Measurements

We run a series of measurements that enable us to identify the roaming setup, infer the network configuration for the 16 MNOs we measure and quantify the end-user performance for the roaming configurations we detect. We run `traceroute`, `dig` for DNS lookups and `curl` for testing data transfers with popular URLs. Furthermore, we complement the analysis with some metadata (e.g., technology, signal strength parameters) we collect from each node.

For each MNO, we measure in parallel the roaming user, the home user and the visited user (see section 5.2 for terminology) through the EURoam scheduler. This way, we are able to capture potential performance penalties that might result, for example, from roaming internationally under a home-routed configuration. We performed measurements using both 3G and 4G networks to evaluate the impact of potentially different configurations for the two radio access technologies.

Next, we describe each measurement test and its resulting dataset in more details.

`traceroute`: We run periodic `traceroute` measurements against all the servers we deploy in each country as measurement responders. Towards each target we repeat the measurements for 10 times on average. The resulting dataset lists the set of IP hops along the data paths from each vantage point towards each measurement responder in each country. Additionally, we collect the public mapped IP address for each vantage point (i.e., the IP endpoint associated with the mobile client as seen from the public Internet).

`dig`: We run the `dig` utility for DNS lookups against a list of 180 target Fully Qualified Domain Names (FQDNs) mapped to advertisement services. We use the independent filter lists from https://filterlists.com to build the list of targets. We focus on ad services since this type of third party services inflate significantly the page load time metrics of web services, as well as impact the web experience of mobile users [118]. Each experiment uses the default DNS server for the tested MNO and queries for the A record associated to each of the target FQDNs. We store the entire output of each `dig` query, including the query time, the DNS server used and the A record retrieved. We repeat the `dig` queries for 2 times for each FQDN from each vantage point, for a total of more than 2,000 queries per round.

`curl`: We run `curl` towards a set of 10 target popular webpages[4] over HTTP1.1/TLS. We repeat the measurements towards each URL for at least 10 times (and we increase this sample if the SIM data quota allows it). We store various metrics, including the download speed, the size of the download, the total time of the test, the time to first byte, the name lookup time (query time) and the handshake time.

[4]We target the following web pages: www.httpvshttps.com, facebook.com/telia/, en.wikipedia.org/wiki/Timeline_of_the_far_future, linkedin.com/company/facebook, www.yahoo.com/movies, instagram.com/leomessi/, google.com/search?q=iPhone+7,youtube.com/watch?v=xGJ5a7uIZ1g, ebay.com/globaldeals, nytimes.com, theguardian.com.uk/lifeandstyle.

Table 5.1 PGW details per MNO

MNO	Visited networks		# of PGW	#of tests	First hop distribution(%)
	3G	4G			
O2 DE	9	9	20	657	1; 1; 1; 2; 2; 2; 2; 2; 3; 3; 3; 4; 4; 5; 6; 6; 9; 9; 11; 24
Telekom DE	5	5	4	1424	13; 19; 25; 43
Voda DE	5	6	2	1511	46; 54
Movistar ES	6	6	8	282	4; 5; 5; 7; 8; 21; 22; 28
Orange ES	7	7	3	900	6; 43; 51
Voda ES	5	5	1	1943	100
TIM IT	6	6	4	497	1; 1; 46; 52
Voda IT	5	5	4	759	19; 19; 23; 39
Telenor NO	5	5	3	398	8; 30; 62
Telia NO	5	5	4	379	7; 16; 38; 39
3 SE	7	6	2	828	44; 56
Telenor SE	5	5	2	1362	32; 68
Telia SE	5	5	4	379	7; 16; 38; 39
EE UK	5	5	9	1038	3; 4; 4; 5; 8; 13; 17; 19; 27
Voda UK	5	5	1	503	100

`metadata`: We collect contextual information from the nodes, including the visited network Mobile Country Code (MCC) / Mobile Network Code (MNC) for each roaming SIM and the radio technology. This allows us to verify which visited network each roaming SIM uses as well as to identify and separate the collected data by radio technology.

5.3.2 Roaming configuration

Our initial goal is to determine the roaming setup for each MNO (i.e., whether it used LBO, HR or IHBO). For this, we determine which is the MNO that allocates the public IP address of the roaming SIM. Our results show that *HR was used by all 16 MNOs from all the different roaming locations we capture*. We further corroborate this result by retrieving the first hop replying with a public IP address along the data path from a roaming SIM to each server and identifying the MNO that owns it. We find that the first hop with a public IP address along the path lies in the original home network of each roaming SIM, which is consistent with HR.

We evaluate next the following performance metrics for each roaming SIM, home SIM and visited SIM: the number of hops from vantage point to target measurement server, the number of visited networks we observe for the roaming SIM, the number of home network PGWs that the roaming SIMs reach in comparison with the home network SIMs.

Visited network selection: The metadata we collect during the measurement campaign for each MNO enables us to verify which is the visited network each roaming user camps on in the visited country. In general, we note stability both in 4G roaming and 3G roaming in the selection of the visited network (Table 5.1) from the five locations. The locations that presented more variability were Spain and Italy. We also observe some differences between MNOs. For example, for Telekom DE the LTE visited network chosen by each roaming SIM never changed during the measurement campaign, even when we forced the radio technology handover. This is consistent for all the five roaming locations. For O2 DE, on the other hand, the default LTE visited network did change in time for the SIMs roaming in Italy (3 visited networks), Norway (3 visited networks), and Sweden (2 visited networks). However, it should be noted that the length of the measurement period varies for each MNO, as it is impacted by multiple external factors (e.g., at times some of our measurement responders were affected by power outages or some SIM cards were not connecting to the 4G network due to poor coverage). This may be part of the differences seen between the MNOs.

Traceroutes, number of hops: We analyze our collected traceroute results from the roaming SIMs and compare with the traceroute results we collect from the corresponding home SIM towards the same target server. For all MNOs we find that *the number of hops is the same*. This is consistent with the HR configuration (Fig. 5.1), where the GTP tunnel is defined between the SGW of the visited network and the PGW of the home network. [5]

Traceroutes, infrastructure: By learning the IP addresses of the infrastructure elements along the data path, we are able to infer aspects of the infrastructure deployment strategy of each MNO. In particular, by checking the IP address of the first hop in the path, i.e., the PGW router, (Table 5.1), we find that MNOs have different strategies in terms of their deployments. We note that the first hops have an even distribution on their assignation to mobile users, showing that the MNOs have a similar approach for load balancing in their network. For example, for O2 DE we find 20 different first hops, suggesting that there might be a large number of PGWs deployed in the LTE infrastructure, while for Vodafone UK we see that the same first hop appears on the data path, suggesting that the GTP tunnels of all our roaming users is terminated at a single PGW. We also note that although for

[5]Traceroute for 3 IT did not work in any country to any server.

the majority of MNOs, these hops are configured with private address space, three operators (Telekom DE, Telenor NO and Telenor SE) use public address space for their infrastructure.

We verify that the set of first hops for roaming SIMs is the same as the set we observe from the home SIMs. This suggests that the roaming SIMs do not receive any differential treatment in terms of allocation to the PGWs. This is consistent for all MNOs we measure. Furthermore, when checking the 3G data paths we find that the set of IP addresses we see in 3G is a subset of the set of IP addresses we see in 4G, suggesting that the two functions are co-located in the same PGW[119]. We also check the time when the first IP was used. We do not find any evidence of dependency between the IP usage and the time. We further contacted 3 MNOs and the information they provided about their network confirms our findings.

5.3.3 Home-Routed Roaming: Implications

Delay Implications: The HR data implies that the roaming user's exit point to the Internet is always in the home network (Fig. 5.1). Thus, the data that the roaming user consumes must always flow through the home network. Depending on the location of the other communication endpoint, this translates into a potential delay penalty. Fig. 5.5 shows the ECDF of the measured RTT between the roaming SIMs and the targets located in the visited or home networks (red and green respectively). In order to compare the HR with the LBO configuration, we also include the RTT measurements between the visited SIMs against the same targets in the visited or home networks (blue and purple respectively). The RTTs experienced by the visited SIMs serve as estimates of the best RTTs that one could expect with a LBO configuration. We note that the largest delay penalty comes when the roaming user tries to access a server located in the visited country. This is because the packets must go back and forth from the home network. Surprisingly, we note that the HR configuration also impacts the case when the roaming user accesses a target server located in the home network. That is, the GTP tunnel is slower than the native Internet path. In this case, the median value of the delay penalty considering all the MNOs is approximately 17ms. This varies across MNOs and in some cases we observe very low penalties (e.g., just 0.2ms for O2 Germany).

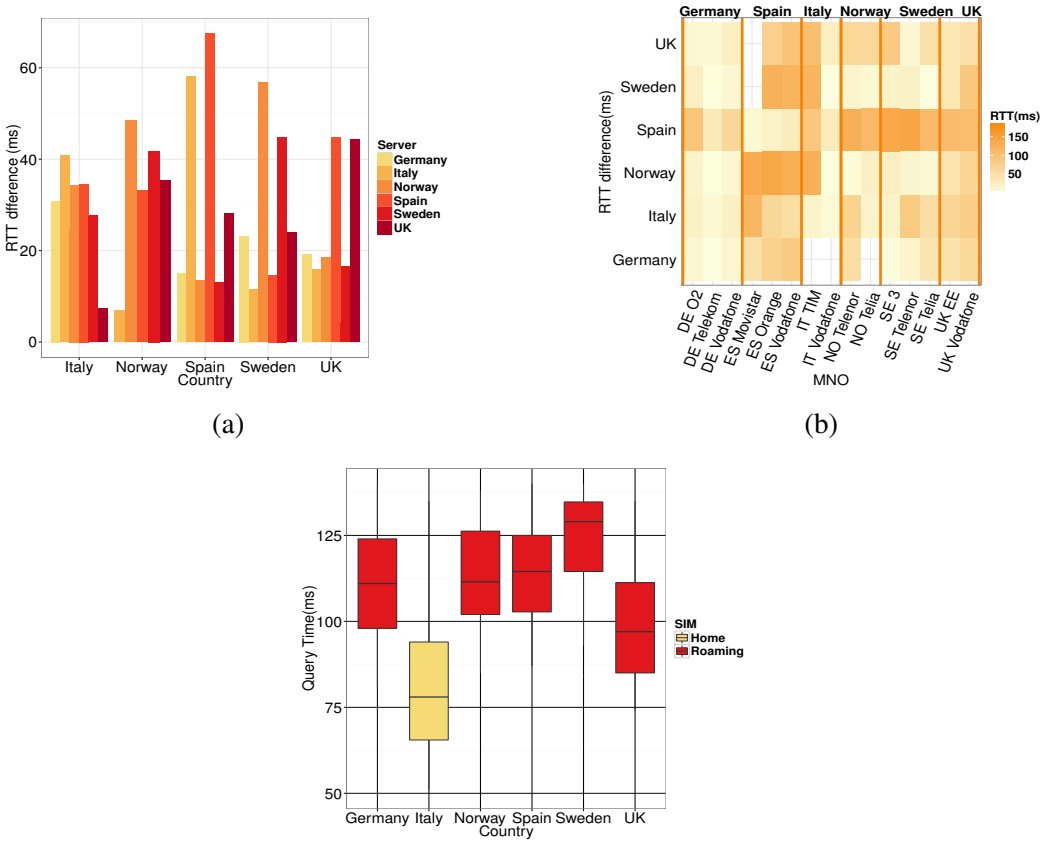


Fig. 5.4 Delay penalty of HR: (a) RTT difference from the visited country to all servers for Vodafone DE; (b) RTT difference per operator; (c) DNS Query time to all FQDNs for TIM IT

We investigate this performance impact further and calculate the estimated delay penalty between LBO and HR when the target is in the visited network. Fig. 5.4a exemplifies these median values for Vodafone Germany. We note that the delay penalty varies widely with the geographical location of the roaming users and the target servers. For example, when a German SIM roams in Spain, the difference in terms of RTT is higher if the server is in the visited country (i.e., Spain) (red curve in Fig. 5.5). If the German SIM roams in Spain or Italy and the target server is in Norway or Sweden the delay penalty of the roaming is less intense, since to go to Norway or Sweden the data path would anyway likely pass through Germany (and this is similar to the delay one would have because of the HR configuration).

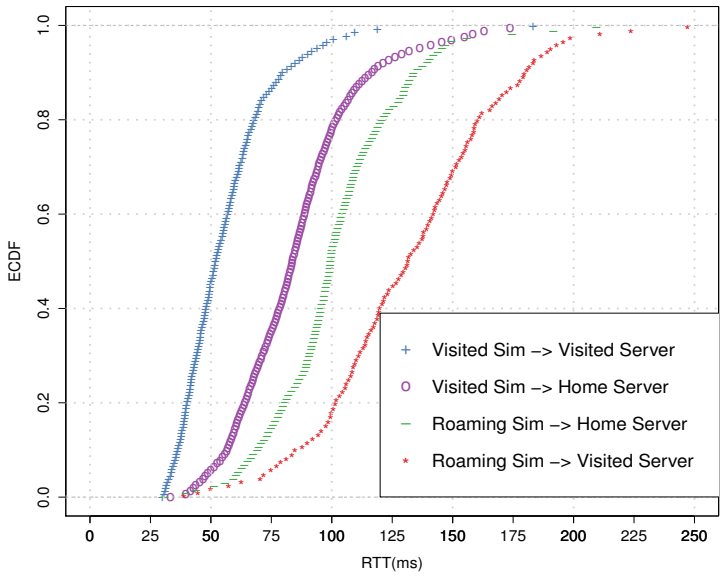


Fig. 5.5 ECDF of the RTT from SIM vantage point to target server

We evaluate the RTT difference between the roaming SIM and the visited SIM towards the same target and we group per MNO. Fig. 5.4b shows the median value of the delay penalty of an MNO (on the `x` axis of the tile plot) while roaming against each the six different servers (on the `y` axis of the tile plot, marked by country). We note that the delay penalty varies as a function of the location of the home country. For example, German SIMs suffer a lower delay penalty, which is potentially due to the advantageous position in the middle of Europe.

DNS Implications:

The results of the `dig` measurements show that the DNS server offered to a roaming user is the same as offered when at home. This is again consistent with the use of HR. We verify whether this translates into an inflated query time for the roaming user. Fig. 5.4c presents the distribution of DNS query times for all the SIMs of TIM IT. We note that for the home user the query time is lower in average than for the other five roaming users. This further translates into implications in terms of CDN replica selection: the roaming user would be likely redirected to CDN content at its home network, and will not be able to access the same content from a local cache.

HTTP performance implications: Similar to the delay and DNS implications, international roaming affects HTTP and HTTPS performance. We quantify this

penalty by considering the handshake time between the SIM and the target web servers. The median value of the handshake time from the visited SIMs towards all the targets we measure is 170ms, while the median value for the roaming SIMs is 230ms. This leads to a delay penalty of approximately 60ms. As in the cases before, some MNOs are affected more by this roaming effect than others.

Takeaway: All operators use HR. This forces all traffic to go through the home country, adding a latency penalty proportional to geographical distance. Interestingly, the GTP tunnel is also slower than the corresponding Internet path. All services are affected.

5.4 VoIP & Content Discrimination

5.4.1 VoIP Call

We focus on three popular Voice over IP (VoIP) applications: Whatsapp [120], Facebook Messenger, and FaceTime [121]. We aim to verify potential traffic differentiation policies (such as blocking or throttling) that may hamper VoIP communications from these applications for a roaming user in comparison to a home user.

Experiment Design: Since it is hard to directly run and control each native mobile application in our platform, we divide the experiment into two parts. In the first part, we check if the MNO allows us to successfully setup an audio/video call by manually running the experiment using regular phones. If successful, we proceed to the second part, where we check for traffic differentiation mechanisms in place.

Towards this end, we made three audio and video calls using each application running on a regular mobile phone connected using an instrumented WLAN access point (AP) in our lab. We recorded packet traces using `tcpdump`, resulting in 18 traces, each with duration between [60,80] s. We verified the call-setup phase, which used a complex mix of TCP, STUN [122], and custom protocols to setup the end-to-end communication. From each trace, we then extracted the actual audio/video streams. In the next step, we created an Docker container with pre-loaded traces, which we replay using `tcpreplay` to properly edit them so that packets are directed toward dedicated receivers in our premises. All applications run SRTP [123] on

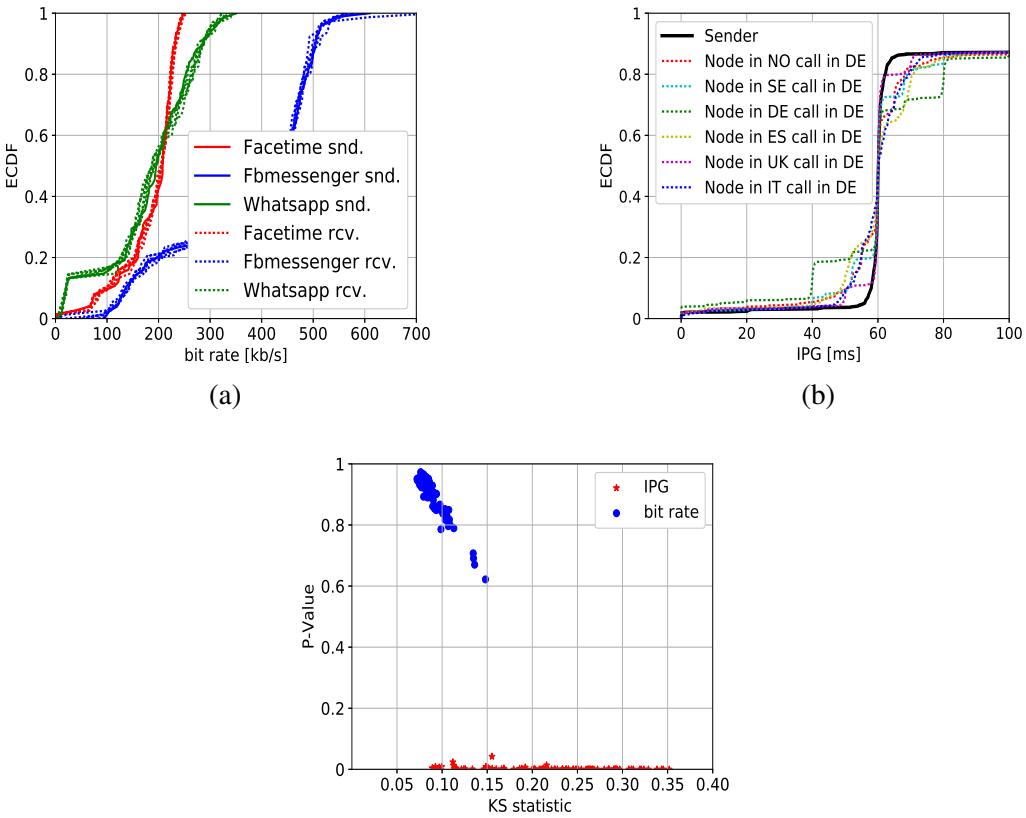


Fig. 5.6 A sample of VoIP results and statistical similarity of bitrate and IPG: (a) Bitrate for operator O2 DE; (b) IPG for operator O2 DE; (c) KS and P-Value test.

top of UDP. This makes it easy to replay the original packet timing, payload, and updated source and destination IP addresses.

The dedicated server in each country acts as a UDP receiver with a custom signaling TCP connection providing the status of the node (such as Visited Network, node identifier, metadata, experiment type, etc.) and experiment association on the receiver side. Each test sequentially replays each pre-recorded trace with two receivers: a call to a destination in the home country, and a call to a destination in the visited country. For each call, we record packets on both the client and server side. Later, we post-process the `pcap` traces to check for eventual traffic differentiation.

Results: First, we verified that *all operators allowed users (even when roaming) freely make VoIP calls*. Then, we consider eventual traffic differentiation. We focus

on packet loss, instantaneous bit rate, and IPG (the time difference between two consecutive packets), all well-known real-time VoIP applications. Results show that the packet loss is always less then 1% in all experiments. As such, we conclude that operators did not introduce any artificial packet loss during our tests.

We went to study throughput and compare the per-second bit rate, as seen at the sender and at the receiver side. In more details, Fig. 5.6a presents the ECDF of the per second bit rate for all three applications. Solid (dashed) lines present the bit rate at the sender (receiver) side, when calling a receiver in home or visited country (operator O2 DE). Notice that applications use different audio/video codec combinations with different requirements in terms of bit rate. However, we no differences are observed with respect to throughput.

At last, we study IPG as shown in Fig. 5.6b. We depict the case of the Facetime application on O2 DE operator in roaming, when calling a receiver in Germany. Notice the periodic 60 ms long IPG typical of low rate audio codecs used by modern VoIP applications. In this experiment we observe some differences when comparing measurement at the sender (solid line) and receiver (dashed lines – one from each visited country) side. That is, some packets get compressed (smaller IPG) while other gets separated (larger IPG). This happens in all experiments, with all operators, also when the sender is in its home country. We ascribe this to the 3G/4G access mechanisms that modulate IPG. Given the IPG is in any case bounded to less than 80 ms, we conclude that this would not hamper voice quality. As such, IPG variations are easily absorbed by the receiver playout buffer [124].

While QoS in terms of IPG and throughput are good, the end-to-end delay could be significantly high due to the HR solution. The delay could grow excessively large when two roaming SIMs call each other. The same effect was know in GSM networks, and fixed by Anti-tromboning [125] solutions.

All experiments present very similar results, and we did not find any evidence of traffic manipulation. We summarize finding by using statistical approaches. We opted for well-known KS Test [126] and *P*-Value [127], two non-parametric tests to determine if two ECDFs differ. In a nutshell, we compare the ECDF at the sender (our reference) with the one observed at the receiver side. If they are statistically similar, KS would be close to 0 while *P*-Value would be close to 1. On the contrary, if the two distribution are significantly different, KS would be greater than 0, and *P*-Value close to 0. Fig. 5.6c shows the scatter plot of the (KS, *P*-Value) points, for all

experiments. Results confirm that the per-second receiver throughput is statistically identical with the sender throughput in all experiments. Conversely, IPG statistics are affected by the 3G/4G access mechanisms that alter the distribution (albeit not impairing the VoIP quality).

Takeaway: We did not observed any traffic differentiation policy. However the additional delay of HR could impair real time communication. This is an old issue (typically referred as tromboning) which has been solved in GSM networks, but is back in 3G/4G VoIP.

5.4.2 Content Discrimination

In this section, we aim to evaluate the availability of content when roaming, in particular how operators filter website content and apply geographical restrictions.

Experiment Design The Open Observatory of Network Interference (OONI) [128] provides software tests for detecting censorship, surveillance and traffic manipulation in the Internet, using open source software. We ran two measurement campaigns to detect network interference in home and roaming scenarios, one geared at website restrictions and another at geo-discrimination of content.

The `ooniprobe` web connectivity test [129] performs the following steps over both the network of interest (tested network, using both home and roaming SIMs) and the Tor network [130]: resolver identification, DNS lookup, TCP connect and HTTP GET requests. Differences in the results for the two networks are indicative of some manipulations in the tested network. Ooniprobe performs DNS queries to disclose the IP endpoint of the DNS resolver in the tested network, and records the response, alongside the response using Google's DNS resolver. A TCP connection on port 80 (or port 443 for URLs that support TLS) is attempted using the list of IP endpoints identified in the responses. HTTP GET requests are then sent towards a list of URLs over both the tested network and over the Tor network and the responses are recorded. The results are made available to the public via the OONI API[6]. These results were then analyzed to identify any interference. `ooniprobe` provides results for HTTP header differences, HTTP body length differences and DNS consistency, which we categorized into a "network manipulation profile" for each network tested.

[6]https://api.ooni.io/

Table 5.2 Interference profile match Home vs. Roaming

	UK	NO	SE	DE	IT	ES
Vodafone UK	100%	98%	99%	98%	99%	86%
Telenor NO	99%	100%	98%	98%	98%	97%
Telia SE	99%	97%	100%	99%	99%	99%
Vodafone DE	98%	97%	99%	100%	98%	97%
TIM IT	98%	98%	98%	97%	100%	97%
Orange ES	99%	98%	98%	N/A	97%	100%

Table 5.3 Interference profile match for geo-restricted websites, Roaming vs. Visited

	UK	NO	SE	DE	IT	ES
Orange ES	N/A	85%	83%	N/A	89%	100%
EE UK	100%	86%	89%	80%	87%	88%
TIM IT	N/A	85%	83%	85%	100%	87%
O2 DE	87%	89%	82%	100%	N/A	84%
Telia SE	N/A	88%	100%	92%	90%	89%
Telenor NO	93	100%	94%	90%	88%	90 %

We tested 50 randomly selected websites from `ooniprobe`'s default global censorship list [131]. For the second set of measurements we provided a list of 15 websites known to be available locally in the tested countries, but geo-restricted abroad due to , e.g., DRM policies. The HTTP responses were searched for known geo-restriction indicators and warnings, as well as signs of manipulation.

Results: We created a network interference profile for each measurement, containing the names of blocked websites and the HTTP body responses. If the HTTP response was unavailable due to censorship, we record the blocking method reported by ooniprobe. The home country measurement served as the baseline profile against which we compared the roaming profiles for that MNO.

We tested a mixture of dynamic and static websites and conclude no content discrimination for a match of 95% and above between the profiles. Anything below this threshold was investigated in more detail. The results as shown in Table 5.2 are consistent with HR for all MNOs, with the exception of one case for Vodafone UK, where 3 websites blocked by DNS in the home country are instead accessible in Spain.

The findings are similar for the geo-restricted content tests, which is also consistent with HR: content available in a user's home country remains available when roaming. To discern this, we searched the body of the websites for known content indicating geo-restriction. In Table 5.3, we present the profile match between the roaming profile and that of the visited network, for geo-restricted content. Where the corresponding SIM for the visited network was not available at the time of measurement, or we failed to collect the results, we display N/A in the table. The percentages do not indicate substantial differences between the content policy of the two networks. Further investigation shows websites alert the user on the restriction only when attempting to retrieve content, whilst loading the page fully otherwise. We found 10 out of 50 websites in our dataset to have this behavior.

Takeaway: HR keeps the same home policies and preferences also while abroad. The user can access home services, but might fail to access local services and vice versa. As a benefit, language preferences are also kept. This might have legal implications, e.g., being able to access content that would be illegal in visited country.

5.5 Related Work

International roaming has received little coverage in terms of large measurement studies, potentially because of the high costs and coordination efforts associated with running such a campaign. Vallina et al. [43] has leveraged crowdsourced measurements and focused only on national roaming agreements between MNOs. Our work presents an extensive measurement study to understand the international roaming ecosystem in Europe since the introduction of the "Roam like Home" initiative.

There has been recent studies focusing on mobile network characterization and performance. For instance, while Huang et al. [132] study LTE network characteristics in a cellular operator in the US, Safari et al. [133] show performance measurement in mobile networks are much more complex than wired networks, due to the different network configurations such as the presence of NATs or PEP, which do vary over time. Kaup et al. [63] run a crowdsourcing campaign to measure RTT and throughput towards popular websites in Germany. They used the dataset to show that the association of a mobile endpoint to the PoP within the operator network

has influence on network performance. Marquez et al. [119] study spatio-temporal traffic characteristics of applications used by 30M Orange subscribers showing temporal patterns differ by application services. Authors in [134] present a mobile app and a mechanism for identifying traffic differentiation for arbitrary applications in the mobile networks. Their results show differentiation tends to affect TCP traffic. Ververis et al. [135] surveys content filtering for a mixture of broadband and cellular ISPs, and finds a lack of transparency around the policies they implement, as well as outdated and poorly implemented blacklists. In our work, we not only focus on network performance of roaming infrastructure, but also identify possible traffic differentiation for particular applications and content discrimination and geo-restriction for users in international roaming.

5.6 Conclusions

While roaming internationally, different network configuration options can affect performance of various applications for the end user. In practice, though there are three possible solutions (i.e., HR, LBO or IHBO), we find that HR is the norm. This comes with performance penalties on the roaming user, who experiences increased delay and appears to the public Internet as being connected in the home country. This has further implications in the selection of CDN server replica when roaming abroad, because the mobile user will access a server in the home network rather than one close to their location. However, in the same time, the roaming user is still able to access (in majority of cases) the geo-restricted services from the home country in its native language.

We put these results in perspective while trying to also speculate on the commercial implications of the 'Roam like Home' initiative. As regulation reduces the ability of MNOs to compete on price, the subscribers' quality of experience will potentially become a key factor in choosing a provider. The subscribers will increasingly start to compare the roaming experience to the home experience. Thus, an expectation of high quality, always-on services in a visited network follows and if a home network fails to deliver in the visited network, the risk of churn increases. To this end, LBO is a natural step for an IP-based service, and could offer lower operational cost, and cheaper tariffs for data, while at the same time we have shown this can eliminate delay and potentially increase capacity for some traffic (dependent

on the destination). Although LBO relies on access to the infrastructure of the visited network which can have implications on service control and charging, offering this could act in the advantage of the first operators to provide the service. Furthermore, in some cases, under the "Roam like Home" paradigm, some users may purchase SIMs from abroad to use in their country under permanent roaming conditions.

Chapter 6

Deadline-Constrained Content Upload from Multihomed Devices

6.1 Introduction

Wireless technologies such as WiFi, 3G, 4G, and soon-to-come 5G, provide access capacities up to hundreds of Mb/s. Multihomed devices are commonly available, offering the chance to transmit over different technologies and networks at the same time. Yet, there are scenarios in which the amount of data being produced and consumed challenges the bandwidth offered by wireless networks.

In this chapter, we look at one of those scenarios. Our interest is motivated and inspired by the real needs of public transport operators. Public transport vehicles (like buses or trains) are equipped with multiple MBB [1] interfaces, several onboard security cameras record videos. Those must be uploaded to a security center where an operator occasionally requests to watch selected portions of the videos. In this scenario, continuous real-time video uploading is too expensive. Even if current MBB networks can offer capacities up to 100 Mb/s, the number of vehicles and videos, the limited data quota, the performance variability along the route, and the need to check only parts of the videos, call for ingenious upload strategies. Hence, videos are stored onboard, and, only when an alarm is triggered, the security operator on duty requests the specific portion of the video that must be uploaded before a specified short deadline. The deadline normally is of the order of a few minutes, depending on the urgency of the incident. For instance, pickpocketing events can wait

till the vehicle returns to deposit. Instead, in case of health problems of passengers, the security officer needs to access the video within a short time, after obtaining all the required authorizations.

We model this problem as the scheduling of content upload from multihomed mobile devices, where the content must be delivered within a given deadline, while the cost must be minimized. The cost associated with each interface is defined according to the nature of the problem. For example cost can correspond to tariffs, energy consumption, data quota, or system load. In our problem definition, cost is related to the monetary cost of the data transmission on each technology.

Our problem differs from the classic problem of content upload using multihomed nodes [136, 137], where upload delay has typically to be minimized, i.e., throughput maximized. Also, no real time constraint exists in our case, thus making our problem different from video streaming, and somehow similar to a delay tolerant scenario, albeit the hard deadline for delivery of the entire content (rather than individual packets) must be met [138].

We assume that the mobile node is equipped with several MBB interfaces, with different technologies, e.g., cheap but occasionally available WiFi, more ubiquitous, but more expensive, 3G, 4G, and soon-to-come 5G subscriptions, possibly offered by different operators. The system has to decide i) which interface(s) to use, ii) when to upload from such interface(s), and iii) at which rate to upload (if there is available bandwidth). Our goal is to minimize the total cost of the upload, while meeting the deadline. A greedy solution that immediately starts uploading from all interfaces minimizes the upload time, ignoring opportunities for cheap interfaces to become available in the future, thus increasing upload cost. A trade-off clearly exists between minimizing the total transmission cost or minimizing the upload completion time.

In this chapter, we propose and analyze a family of adaptive schedulers that require only a very coarse knowledge of the available bandwidth on wireless interfaces. We extend our work in [139] by defining a more refined scheduler, and evaluating our solution using a larger and recently collected dataset to evaluate the dynamic algorithm proposed in [140]. In addition, We discuss how to carefully tune the dynamic algorithm parameters, and we provide a more extensive evaluation. Finally, we implement and test the algorithms in a real testbed, provided by the MONROE platform [117].

The main contributions of this chapter are:

• Devising mathematical formulations of the deadline constrained content upload problem from multihomed terminals, under different assumptions.

• Reporting extensive evaluations of the proposed solutions, based on trace-driven simulations using recently collected traces.

• Designing, implementing, testing, and evaluating a real implementation of the proposed dynamic algorithm on deployed mobile multihomed nodes.

The rest of this chapter is structured as follows. We first overview related works and we position our work with respect to state-of-the-art solutions for similar problems (Section 6.2). Then, we report on the collection of traces of content upload data rates from mobile multihomed terminals, showing the unpredictability of short-term variations in available bandwidth (Section 6.3) to gain insight about the trade-off between cost and delivery time over wireless channels. Next, we formulate and solve an idealized version of the problem, where an oracle has perfect knowledge of the upload rate on each interface at each time. The oracle can then schedule the upload in those time slots when cheap connectivity is (expected to be) available, thus minimizing total cost (Section 6.4.1). We also introduce three simple greedy heuristics, to show the effectiveness of intuitive approaches to solve this problem (Section 6.4.2). Then, we formulate the video upload problem as a centralized scheduling problem, where the upload rates of the available interfaces are random variables with known distribution. Solving such problem is computationally impractical (Sec. 6.4.3). Thus, we aim for a practical solution that requires only a coarse knowledge of the available bandwidth, and we design online, adaptive schedulers to explore the trade-off between cost and delivery time (Section 6.4.4). Afterwards, we test the proposed algorithms in the real MBB platform provided by the MONROE project (Section 6.6). Finally, we conclude the chapter (Section 6.7).

6.2 Related Work

Mobile devices allow users to connect to multiple wireless networks with possibly different technologies [141–143], obtaining throughput values which depend on the terminal position, the network coverage, the traffic load, the weather conditions, etc.

This makes the problem of scheduling transmissions over multihomed [144] wireless interfaces both relevant and challenging. Several authors already published works which are close to what we discuss in this thesis. For example, Higgins et al. [145] were among the first to face this problem. They tackle a problem in this domain, and propose Intentional Networking, that lets the application choose opportunistically the interfaces, based on a label that expresses the application requirements. They however target different problems such as real-time communications, with no support for deadline.

We discuss related work by looking at three main dimensions, which correspond to topics which have been widely investigated in the past: predictability of wireless network performance, multipath TCP, delay tolerant networks, and deadline scheduling.

• **Predictability of MBB performance:** The performance of wireless network services under uncertain network availability has been previously investigated by several authors. Deng et al. [146] investigate the characterization of multihomed systems considering WiFi vs. LTE in a controlled experiment. They show that LTE can provide better performance than WiFi, also exhibiting large variability on both short and long time scales. Rahmati et al. [147] present a technique for estimating and learning the WiFi network conditions from a fixed node. Rathnayake et al. [148] demonstrate how a prediction engine can be capable of forecasting future network and bandwidth availability, and propose a utility-based scheduling algorithm which uses the predicted throughput to schedule the data transfer over multiple interfaces from fixed nodes. These works heavily rely on channel performance predictions, and consider scheduling at the packet-level, i.e., choosing which packet to send through which interface, to maximize the total throughput.

Our work differs from those, since we deal with moving vehicles, and this exacerbates the unpredictability of the network performance, as shown by several authors. For instance, Riiser et al. [149] collected 3G mobile network traces from terminals onboard public transport vehicles around the city of Oslo (Norway). Similarly, Chen et al. [150] measured the throughput of both single-path and multi-path data transport in 3G, 4G, and WiFi networks. In both cases, variability is much higher than for fixed nodes. Lee et al. [151] showed that mobile data offloading through WiFi can reduce the energy consumption of the mobile device, Bychkovsky et al. [152] presented the connectivity characteristic of WiFi for a mobile device traveling in a city. Similarly,

Safari et al. [133] ran large-scale download measurements in MBB networks. They show MBB networks are much more complex than wired networks, because of many factors which clutter the picture.

Given this difficulty in predicting the characteristics of MBB networks, we collected traces, and we used them to run trace-driven evaluations in realistic scenarios.

• **Multi-path TCP:** A number of recent works focus on multi-path TCP (MPTCP) [136, 153], and look at the design of packet schedulers and congestion control algorithms. The goal of the authors normally is to maximize throughput, or equivalently to minimize upload time (rather than to minimize the total cost of uploading a given content within a specified deadline, as we do). Nikravesh et al. [137] thoroughly investigate the performance gains and the costs of mobile MPTCP by means of traffic measurements, and present the MPFLEX software architecture. Wu et al. [154] propose a framework for video streaming, but do not consider the cost associated with interfaces or a deadline for video upload, see also [155–157]. Lim et al. [158, 159] introduce an energy-aware variant of MPTCP which aims to reduce energy consumption with respect to standard MPTCP. They however consider download transfers, do not consider any deadline, which makes their work different from ours. Han et al. [160] show the fact that MPTCP implies unnecessary use of cellular interfaces in case of available bandwidth on WiFi in multihomed system. They present MP-DASH, a network interface preference-aware multi-path framework for DASH video streaming. Its performance is highly dependent on the choice of the DASH algorithm. This work is different from ours because it focuses on real-time streaming solution.

• **Delay tolerant networks:** Delay Tolerant Network (DTN) solutions for content upload try to find the way to deliver the content by maximizing the device-to-device transmission [161–163] or maximizing the use of WiFi [164, 165]. The data delivery has no deadline, and the main problem is the creation of the time-varying network topology to guarantee the delivery. Yetim et al. [164] illustrate the benefit of the delay tolerant approach to save energy by sending more data over WiFi interface. However, their approach does not support change in cost and deadline. On the contrary, we rely on MBB to offer connectivity with associated cost, and devise approaches to use interfaces so as to minimize cost while delivering the content before the deadline.

• **Deadline scheduling:** Previous works did consider scheduling under a fixed deadline, but they assumed that network performance is perfectly known. Zaharia et al. [166] presented an optimal scheduler over multiple network interfaces, and proposed approximations which can be implemented with limited resources in mobile phones, or PDAs. They assumed the cost and bandwidth of each interface to be constant. Moo-Ryong et al. [167] also proposed an algorithm for video upload from smartphones with two MBB interfaces. They focus on energy-delay trade-off. These works differ from ours, since they assume MBB interface availability and capacity are known (which we consider not realistic). We do not assume any a priori knowledge of available capacity.

6.3 Characterization of Mobile Traces

We first present some recent mobile traces that we collected from vehicles, with the dual purpose to show how unpredictable the available bandwidth is, and to run realistic performance evaluation using trace-driven simulations; indeed, a credible evaluation of the proposed algorithms calls for realistic data about available upload bandwidth from public transport vehicles. Previous studies collected traces of MBB network data rates, e.g., Chen et al. [150] and Riiser et al. [149]. However, these traces are not very recent, hence they exhibit lower bandwidth values than the ones that we commonly experience over today's wireless networks. Lutu et al. [168] collected a large set of recent traces for MBB from public transport vehicles. However, they primarily focus on the transfer of relatively short files (4 MB) that is less likely to effectively utilize the available bandwidth, and to represent its variations over a longer time scale (minutes vs. seconds). We thus resolved to collect our own traces by using mobile terminals onboard private and public vehicles, or carried by walking users.

6.3.1 Trace Collection Methodology

All traces were collected in the city of Torino in Italy in 2016, and refer to three technologies (WiFi, 3G, and 4G), and different mobile network operators. During trace collection, the MBB networks were in normal operating conditions (and unaware of

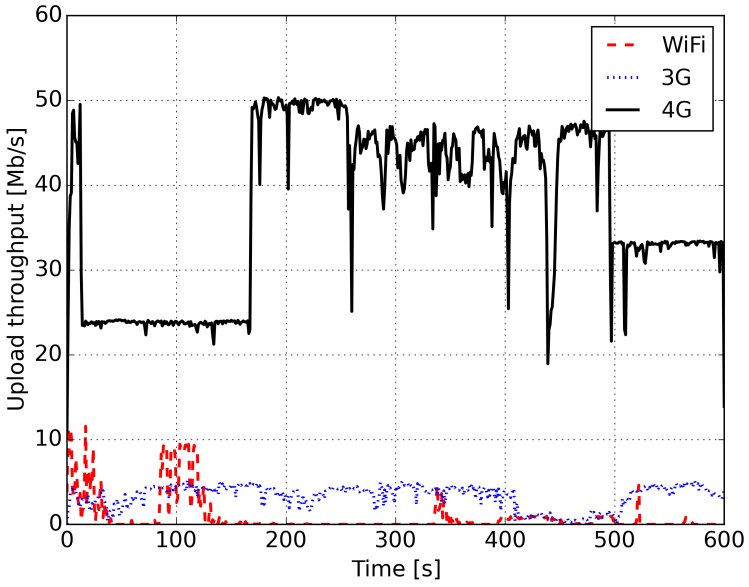


Fig. 6.1 A sample of collected traces for each technology

our tests). Our terminals (both Android and iOS smartphones) accessed the mobile networks to upload data to a server on campus. We used both TCP and UDP.

We used a hybrid method in the trace collection process: during each experiment, the mobile terminal runs `iperf2` [1] in the upload direction for 600 seconds while `tcpdump` [2] captures packets at the server. Using the packet trace, we compute the throughput in each second of the experiment. The number of repetitions of active measurements is critical to make sure that enough samples are collected for a sound estimation of the distribution of the throughput of each technology. It is important to note that repetitions cover different times of the day and different days of the week. We repeated the experiment on the same driving routes for at least 5 times, during different days.

In total, we collected 40 traces for each of the three different mobile network operators in Italy (namely TIM, Wind, and Vodafone), with the objective of obtaining multiple samples of the upload throughput in MBB networks. Traces are collected in mobile scenario, the speed of vehicles were in range of [0,70]km/h, most of them collected in three routes with length [2,5]km. For WiFi, we considered the open

[1] https://iperf.fr/iperf-doc.php
[2] http://www.tcpdump.org/

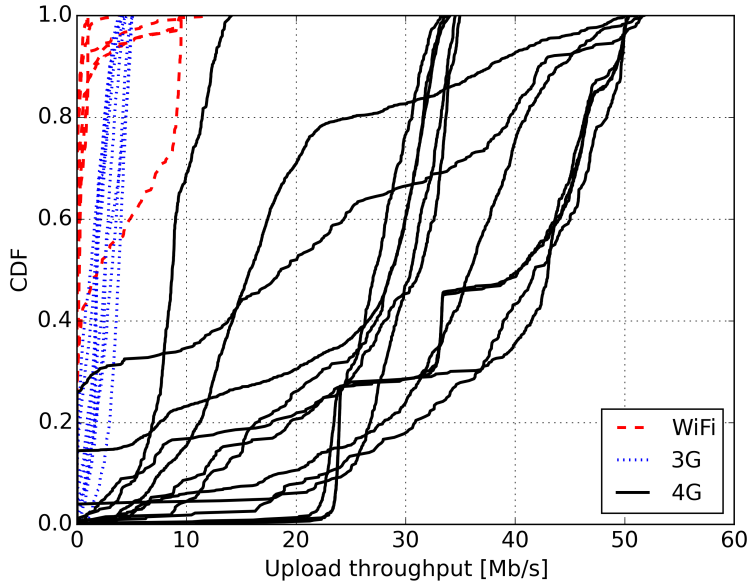


Fig. 6.2 CDF of throughput for multiple traces and technologies

Table 6.1 Throughput statistics

interface	mean	standard deviation	80-th percentile	max	min
WiFi	0.77 Mb/s	2.06	0.57	11.56	0
3G	2.23 Mb/s	1.29	3.44	5.23	0
4G	26.92 Mb/s	13.50	39.47	51.74	0

WiFi community "WoW-Fi" offered by Fastweb customers that share their ADSL or FTTH home networks via the access gateway.[3] We make the collected traces available for researchers[4].

Table 6.2 $|Th_t - Th_{t-1}|$ statistics

interface	mean	standard deviation	80-th percentile	max	min
WiFi	0.30 Mb/s	0.89	0.24	9.45	0
3G	0.39 Mb/s	0.44	0.63	3.84	0
4G	2.02 Mb/s	2.98	2.93	47.14	0

[3] http://www.fastweb.it/adsl-fibra-ottica/dettagli/wow-fi/. Mobile phones automatically authenticate using IEEE 802.1x with no action needed from the user.

[4] http://tstat.polito.it/traces-MBB-speedtest.shtml

6.3.2 Trace Characterization

We now present evidences of unpredictability of throughput, which makes the scheduling of content upload a challenging task. Fig. 6.1 shows a sample of the temporal evolution (x-axis) of the upload rate (y-axis) for WiFi (red), 3G (blue), and 4G (black) interfaces, collected when using UDP and TCP. For each trace, we compute the upload throughput considering time intervals of 1 second. Starting times of traces have been re-aligned for ease of visualization. Fig. 6.2 presents the ECDF of the per second upload rate for multiple randomly selected traces. Figures 6.1 and 6.2 indicate that WiFi offers upload throughputs which are very variable in time, with a behavior that is almost ON-OFF. Upload rates are limited to less than 10 Mb/s. This is due to the limited coverage of the WiFi network, and to the upload bottleneck of ADSL or FTTH access technologies. The 3G technology provides upload rate values which are invariably lower than 5 Mb/s, with significant short-term variability, but, thanks to the extensive coverage, no long periods of close-to-zero available bandwidth were observed. The behavior using the 4G interfaces exhibits even higher variability, with rates one order of magnitude higher than 3G (up to 50 Mb/s). We do not observe any significant differences when using TCP or UDP, since the unpredictable changes in the rate are mostly due to sudden changes in the access link than to congestion along the path.

Table 6.1 shows, as a summary of the statistics for the three technologies, the average, standard deviation, 80-th percentile, maximum, and minimum of the observed per second upload rate. Table 6.2 reports the same statistics, but considering the absolute difference of throughput in two consecutive time slots. In a nutshell, measurements indicate that it is not realistic to assume the exact value of the future available bandwidth, as also claimed by Nikravesh et al. [169].

6.4 Problem Formulations and Algorithms

Although we just claimed that assuming to know the future available bandwidth is not realistic, we start by considering this case, since it provides a baseline for performance evaluation. As a second step, we present a stochastic formulation of the problem, that assume only the probability distribution of the available bandwidth is

known. Finally, we propose dynamic schedulers that adapt their choices based only on the past values of available bandwidth.

6.4.1 Optimal Solution with Perfect Bandwidth Knowledge

We assume an oracle has perfect knowledge about the bandwidth of all interfaces at all times. We consider time slots, with slots of duration ΔT. The time slot duration is such that the available bandwidth over all interfaces can be assumed constant for one time slot.

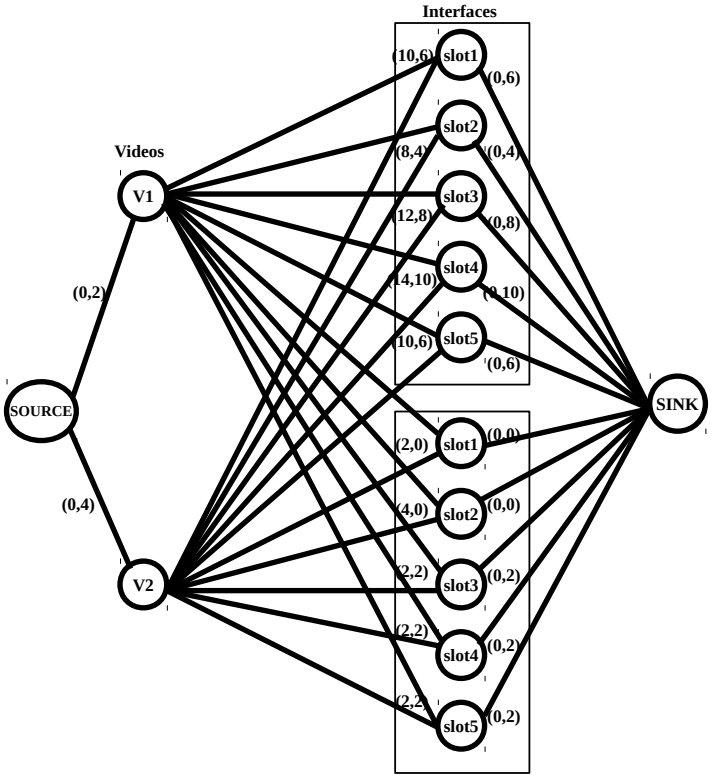


Fig. 6.3 An example to represent the MCFP model with 2 videos, deadline equal to 5 time slots, and 2 interfaces

We model the scheduling problem using a directed graph $G_1 = (N, E)$, where N is the set of nodes and $E = \{(i, j) \mid i, j \in N\}$ is the set of edges. Referring to Fig. 6.3, the leftmost node in G_1 represents the video source, i.e., the vehicle. The second group of nodes represents the video files to be uploaded. Each video $k = 1, \ldots, K$ (2 videos in the example) is of volume V_k, and can be uploaded through different interfaces. The system has T time slots to complete the upload, represented by the third group of nodes. Each node in this group represents a given interface and time slot. For ease of visualization, nodes referring to the same interface (2 interfaces in the example) are grouped in a box. The number of available time slots (5 in the example) represents the deadline to be met. The rightmost node represents the sink, i.e., the server receiving the videos.

Edges in E are labeled by a cost $c_{i,j}$ and a bandwidth $r_{i,j}$. The label of edge (i, j) is denoted $(c_{i,j}, r_{i,j})$. The source node is connected to each video node. Edges exiting from the source node have zero cost, and bandwidth equal to the video file size in bits. Edges from video nodes to time slot and interface nodes are characterized by the cost per bit of using such time slot and interface $(c_{i,j})$, and the maximum flow that can be supported by such time slot and interface $(r_{i,j})$ in bits/s. This model allows videos to have different deadlines. Indeed, each video is connected only to the slots it can use. nodes representing time slots and interfaces are connected to the sink with an edge with zero cost, and bandwidth equal to the time slot bandwidth.

We assume that there is enough bandwidth to successfully upload all videos, and that any interface can be shared between any video at any time slot, i.e., video content is fluid, and can be split arbitrarily.

Minimum Cost Flow Problem Model

We model this problem as a Minimum Cost Flow Problem (MCFP) [170], in which we look for the maximum flow that the network can carry, with the minimum total cost. The objective function in (6.1) presents the total upload cost, which must be minimized. Expression (6.2) forces flow conservation constraints. It states that the sum of incoming flows at all nodes (except source and sink) is equal to the sum of outgoing flows, i.e., flow cannot be created or disappear at intermediate nodes. The flow on every edge is non-negative, and it cannot exceed the rate $r_{i,j}$, see (6.3). Expression (6.4) forces the total flow exiting from the source node to be greater or equal to the sum of all requested videos, i.e., all videos must leave the source.

Table 6.3 Variables definition for MCFP

variable	definition
t	Time slot index $\in [0 \dots T]$
ΔT	Duration of time slot
T	Number of time slots before the deadline
K	Number of videos
I	Number of interfaces
N	Total number of nodes in the graph
$r_{i,j}$	Available bandwidth on edge from node i to node j
$f_{i,j}$	Amount of data scheduled from node i to node j
$c_{i,j}$	Cost associated to edge from node i to node j

$$\textbf{min} \sum_{(i,j)\in E} c_{i,j} f_{i,j} \Delta T \tag{6.1}$$

$$\sum_{(i,j)\in E} f_{i,j} = \sum_{(l,i)\in E} f_{l,i} \;\; \forall i \in Nandi, j, l \neq Source, Sink \tag{6.2}$$

$$0 \leq f_{i,j} \leq r_{i,j} \;\; \forall (i,j) \in E \tag{6.3}$$

$$\sum_{(Source,j)\in E} f_{Source,j} \geq \sum_{i} V_i \tag{6.4}$$

The MCFP problem can be solved using well known and efficient approaches, like the one in [171], with complexity of $O(m \log n(m + n \log n))$ on networks with n nodes and m arcs. As depicted from graph G_1 n and m are equal to $n = K + (I * T) + 2$ and $m = (K+1)(I*T) + K$. In this work, we use the CPLEX [172] solver.

6.4.2 Heuristic Approaches with Full Knowledge

In order to have simpler alternatives for the computation of the (quasi) optimal solution in the case of perfect bandwidth knowledge, we consider three simple and intuitive greedy heuristics:

i) Greedy-in-time (GT) - This algorithm uploads all videos through all interfaces as soon as possible. It minimizes the upload time, greedily uploading as

much data as possible through all available interfaces. The video with closest deadline is transmitted first, as soon as any interface has an available slot to upload (part of) the video.

ii) Greedy-in-rate (GR) - This algorithm sorts time slots according to decreasing transmission rate, and schedules transmission through the highest-rate time slots. If rates are equal, earlier time slots are preferred.

iii) Greedy-in-cost (GC) - This algorithm sorts time slots according to increasing cost, and schedules transmission through the cheapest time slots. If costs are equal, earlier time slots are preferred.

All heuristics stop when expression (6.4) is met, i.e., all videos are uploaded. The first greedy algorithm guarantees that the transfer is completed as soon as possible (this makes it similar to MPTCP solutions), while the second one minimizes the number of time slots to use. Both approaches disregard the upload cost. Only the third algorithm explicitly considers the cost of using different interfaces at different times.

Assume T is the number of time slots, I is the number of interfaces, and K is the number of videos. The GT algorithm only needs the temporal ordering of slots, which is given, so that complexity is $O(1)$. The GC algorithm needs the ordering of time slots according to cost, so that complexity is $O(T * I * K)$ (which depends on the video and the interface), and then according to time. The GR algorithm needs the ordering according to slot bandwidth, and time, so that complexity is $O(T * I)$.

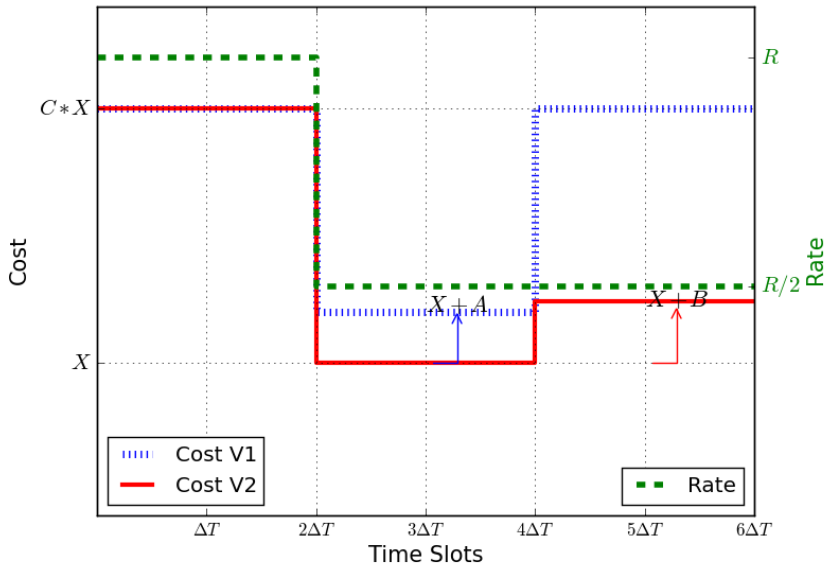


Fig. 6.4 Example to show how greedy approaches can perform worse than the optimal solution

To show that greedy approaches can produce a solution which has suboptimal cost, we use a very simple example with only 1 available interface, and 2 videos to be uploaded, of sizes $V_1 = V_2 = R\Delta T$. The slot costs are presented in Fig. 6.4, with blue dotted and red solid lines for V_1 and V_2, respectively. The costs of V_1 and V_2 are different (e.g., to represent priorities), and change over time (e.g., due to different tariffs at different time of day). This can describe a scenario with congestion-based pricing. Available bandwidth varies according to the green dashed line. The two videos have the same deadline equal to $6\Delta T$. Each video can be uploaded in one time slot with rate equal to R bits/s, or 2 time slots with rate $R/2$. We can easily compute the total cost, considering the greedy heuristics and the optimal solution. By assuming that $A < B$ and $C > 2$, we have:

1. *GT* - The two videos are uploaded in the first two time slots (either one slot per video, or sharing the slot bandwidth), without considering the cost of slots. The total cost is $2CXR\Delta T$.

2. *GR* - The two videos are uploaded in the first two time slots, which have highest rate, without considering the cost of slots. The total cost is $2CXR\Delta T$.

3. *GC* - Since the upload of V_2 has the lowest cost in the third and fourth time slots, which have rate $R/2$, the upload of V_2 is scheduled in those slots. The cost for the upload of V_1 is equal to CX in all slots except the ones that are allocated to the upload of V_2. The first slot is chosen because of the high rate. The total cost is $CXR\Delta T + 2XR\Delta T/2 = (C+1)XR\Delta T$.

4. *Optimal solution* - The solution based on MCFP schedules the upload of V_1 in the third and fourth time slots, and the upload of V_2 in the fifth and sixth time slots. The total cost is $(X+A)R\Delta T + (X+B)R\Delta T = (2X+A+B)R\Delta T$.

We conclude that the total cost for GT and GR is C times higher with respect to the optimal solution. Instead, the cost for GC is $(C+1)/2$ times higher with respect to the optimal solution where $(A+B) \to 0$. By means of this example we have shown that greedy algorithms can generate solutions with possibly much higher cost than the optimal solution, even in very simple cases.

6.4.3 Multistage Stochastic Model

Since assuming the perfect knowledge of the available bandwidth on all interfaces in all time slots is not realistic, we next look at a case with reduced information. We consider the available bandwidth of interface i in slot t, $R_{i,t}$ as a random variable, denoting the realization of $R_{i,t}$ with $r_{i,t}$. From frequent large-scale measurements, it can be possible to estimate the probability distribution of $R_{i,t}$, although this requires great effort. Naturally, the values $r_{i,t}$ are known for past slots, as data are transmitted over the interfaces. Uncertainty in data can be modeled through multistage stochastic models, as follows:

Given the distribution of $R_{i,t}$, we model the scheduling problem using a series of directed graphs $G_2(t) = (N,E)$, where N is the set of nodes and $E = \{(i,j) \mid i,j \in N\}$ is the set of edges. As before, we assume time is slotted, with slot duration ΔT. K videos, each of volume V_k, $k = 1,\cdots,K$, have to be uploaded through I interfaces. The number T of available slots represents the deadline. Edge (i,j) in E is labeled by two values: a cost, and a bandwidth, denoted $c_{ij,t}$ and $r_{ij,t}$ at time t, respectively. Fig. 6.5 illustrates the graph $G_2(t)$ at time $t = 0$.

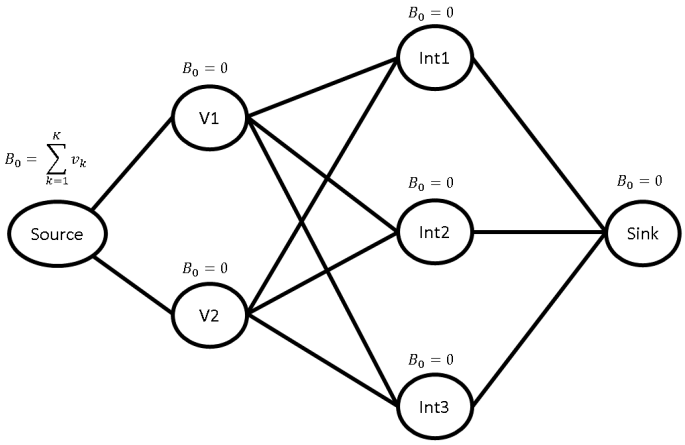


Fig. 6.5 Stochastic model representation at time $t = 0$

A source node is connected to the K nodes representing videos. Edges exiting from the source node have zero cost, and bandwidth equal to the video file size in bits. Each video node is then connected by a directed edge to I nodes, each representing an interface. These edges are labeled by the cost per bit of using an interface $(c_{ij,t})$ at time t, and the maximum bandwidth that can be supported at time slot t $(r_{ij,t})$ in bit/s.

Last, interface nodes are connected to a sink node, with an edge with zero cost, and bandwidth equal to infinity. Only after making a decision for the data to send over an interface at time t, the actual bandwidth $r_{ij,t}$ is known. The quantity $b_{i,t}$ represents the amount data in buffer at node i because the bandwidth was lower than expected at previous time slots. Therefore, the buffer occupancy evolves over time based on decisions made in previous time slots.

The quantity $x_{ij,t}$ represents the amount of successfully transmitted data at time slot t on edge ij. Table 6.4 summarizes the variables used to formulate the problem. Using the graph $G_2(t)$, the problem of minimizing the total cost C to deliver all videos can be solved as a Multistage Minimum Cost Flow Problem (MSMCFP), in which we look for the maximum flow that the network can carry, with the minimum total cost.

The objective function in (6.5) presents the expected value of the total cost over the deadline, which must be minimized, while (6.6) forces the real flow passing from node i to node j to be the minimum between what is scheduled $f_{ij,t}$ and the actual bandwidth of the interface $r_{ij,t}$ at time t. Expression (6.7) states that flow cannot

Table 6.4 Variables definition for MSMCFP model

variable	definition
t	Time slot index $\in [0 \dots T]$
T	Number of time slots
K	Number of videos
I	Number of interfaces
$b_{i,t}$	Amount of data in buffer at node i at time t
$r_{ij,t}$	Available rate from node i to node j at time t
$f_{ij,t}$	Scheduled data from node i to node j at time t
$x_{ij,t}$	Transmitted data from node i to node j at time t

appear/disappear at intermediate nodes at any time, the data being either transmitted or stored in buffers. The flow on every edge must be non-negative, and it cannot exceed rate $r_{ij,t}$, as dictated by (6.8). Expressions (6.9) and (6.10) force the total flow exiting from the source node and entering in the sink node to be equal to the sum of all video sizes. Expressions (6.11), (6.12), and (6.13) indicate the state of buffers at any time t.

$$\min C = \sum_{t=0}^{T} \mathbb{E}\left\{ \sum_{(ij)\in E} x_{ij,t} c_{ij,t} \Delta T \right\} \tag{6.5}$$

$$x_{ij,t} = \min(f_{ij,t}, r_{ij,t}) \;\; \forall ij,t \tag{6.6}$$

$$b_{i,t+1} = \max\left(b_{i,t} + \sum_{j:(j,i)\in E} x_{ji,t} - \sum_{j:(i,j)\in E} x_{ij,t}, 0 \right) \forall ij,t \tag{6.7}$$

$$0 \le f_{ij,t} \le r_{ij,t} \qquad \forall (ij) \in E,\, t \in \{0 \dots T\} \tag{6.8}$$

$$\sum_{t=0}^{T} \sum_{j:(Source,j)\in E} x_{Source,j,t} = \sum_{i=1}^{K} V_i \tag{6.9}$$

$$\sum_{t=0}^{T} \sum_{i:(i,Sink)\in E} x_{i,Sink,t} = \sum_{i=1}^{K} V_i \tag{6.10}$$

$$b_{i,T} = 0 \; \forall i, i \neq Sink \text{ and } b_{Sink,T} = \sum_{i=1}^{K} V_i \tag{6.11}$$

$$b_{i,0} = 0 \; \forall i, i \neq Source \; and \; b_{Source,T} = \sum_{i=1}^{K} V_i \tag{6.12}$$

$$b_{i,t} \geq 0 \; \forall i, t \tag{6.13}$$

To solve the MSMCFP problem, we need to obtain accurate estimates of the probability distribution for the bandwidth of each interface. This requires, as we already noted in Section 6.3, large-scale measurements at different times in all possible places, which is unrealistic. In addition, no off-the-shelf solver is available to solve multistage stochastic problems; the problem can be solved only by searching through all possible scenarios, which requires the exhaustive exploration of a tree of realizations of depth T, with nodes of degree I, i.e., a complexity $O(I^T)$. This means that this more realistic option, which only assumes the availability of probabilistic information about the bandwidth available on interfaces, is not viable because of the solution complexity. This means that the only feasible option to solve the video upload problem is to design adaptive heuristics.

6.4.4 Dynamic Heuristic

Given the complexity of solving the MSMCFP problem, we designed an adaptive algorithm that is inspired by schedulers for P2P video streaming proposed by Magharei et al. [173]. The dynamic of variations for individual connection as well as the design goals in our problem are however different from those in [173]. We only assume the knowledge of the long-term average throughput of each interface. This information serves as a reference to assess the feasibility and the pace of progress for meeting the specified deadline.

We consider slotted time, where ΔT denotes the duration of a single slot. At the beginning of each slot, the scheduler computes the amount of data to transmit on each interface using the observed throughput in recent past slots. It updates the expected rate on each interface based on the overall pace of upload progress during the recent slots, and schedules the transmission of a portion of the data, giving preference to cheaper interfaces. During the slot, data is transmitted according to the actual network state. At the end of the slot, the scheduler checks whether the amount of transmitted data is smaller than expected. If this happens, the unsent data, denoted by $LeftB$, is greater than zero. $LeftB$ is the sum of the amounts of data remains in all the interfaces' buffers at the end of the time slot. When $LeftB < 0$, the system is

behind the expected schedule, and the scheduler needs to recover in the future. We consider two policies for recovery: i) aggressively recovering during the next slot, ii) conservatively (i.e., optimistically) recovering across all the remaining slots before the deadline. Let t be the current time slot. B_t represents the expected data rate at which the system should transmit during slot t. At the upload start, we estimate $B_0 = V/T$, being $V = \sum_{k=1}^{K} v_k$ the total data volume size, and T the deadline. At the end of each time slot, the system computes $LeftB$, the total amount of scheduled data that was not possible to transmit due to a lack of bandwidth in that slot.

To help the scheduler, each interface i maintains the expected rate $\widehat{r}_{ti}$ based on the actual transmission bandwidth r_{ti}. If the interface i was active in period t ($r_{ti} > 0$) and congested ($LeftB_i > 0$), $\widehat{r}_{ti}$ is updated using an Exponentially Weighted Moving Average (EWMA) algorithm with α coefficient:

$$\widehat{r}_{t+1i} = \begin{cases} \alpha\widehat{r}_{ti} + (1-\alpha)r_{ti} \text{ if } r_{ti} > 0 \textit{ and } LeftB_i > 0 \\ Max(\widehat{r}_{ti}, r_{ti}) \text{ if } r_{ti} > 0 \textit{ and } LeftB_i = 0 \\ \widehat{r}_{ti} \text{ otherwise} \end{cases} \tag{6.14}$$

The rationale of expression (6.14) is to avoid the estimated bandwidth to converge to small values when an interface is not being used, or used at a rate lower than the maximum available bandwidth, i.e., when the interface bandwidth is not fully utilized. Indeed, data is transmitted at the expected rate and the actual available bandwidth of the interface is unknown. Thus, we avoid decreasing the estimated rate of those interfaces that are not fully utilized. This happens because the algorithm is not greedy, and interfaces are partially used in a demand-driven fashion.

Algorithm 1 Adaptive Scheduler

1: **procedure** ADAPTIVESCHEDULER($\alpha, \beta, policy$)
2: $\widehat{r}_{0i} \leftarrow$ *Interface average rates*
3: $B_0 \leftarrow V/T$ # minimum rate to meet the deadline at time t=0
4: **for** $(t = 0; t < T$ && $V > 0; t++)$ **do**
5: SortInterfaceByCost()
6: **procedure** PUSH DATA TO BUFFERS
7: **for** $(i = 1; i \leq I$ && $V_t > 0; i++)$ **do**
8: **if** $cost(i) < maxcost$ **then**
9: $V_t = (\beta + 1)B_t\Delta T$
10: **else**
11: $V_t = B_t\Delta T$
12: $V_{ti} = min(V_t, \widehat{r}_{ti}\Delta T)$
13: $V_t = max(V_t - V_{ti}, 0)$
14: UploadAndWaitForSlotEnd()
15: **procedure** CHECK DATA IN BUFFERS
16: $V = V - \sum_i^I r_{ti}\Delta T$
17: **for** $(i = 1; i \leq I; i++)$ **do**
18: $LeftB_i = V_{ti} - r_{ti}\Delta T$
19: **if** $r_{ti} > 0$ && $LeftB_i > 0$ **then**
20: $\widehat{r}_{t+1i} \leftarrow \alpha\widehat{r}_{ti} + (1-\alpha)r_{ti}$
21: **else if** $r_{ti} > 0$ && $LeftB_i = 0$ **then**
22: $\widehat{r}_{t+1i} \leftarrow Max(\widehat{r}_{ti}, r_{ti})$
23: **else**
24: $\widehat{r}_{t+1i} \leftarrow \widehat{r}_{ti}$
25: $LeftB = \sum_i^I LeftB_i$
26: **if** $LeftB > 0$ **then**
27: **if** policy == Aggressive **then**
28: # update minimum rate to meet the deadline at time t=t+1
29: $B_{t+1} = B_0 + LeftB$
30: **else**
31: $B_{t+1} = B_t + LeftB/(T-t)$

Algorithm 1 presents the pseudo code for adaptive scheduling. After initialization, the algorithm loops over time slots until the deadline is reached, or all the data

have been uploaded (line 4). To minimize cost, interfaces are prioritized for data transmission from the least to the most expensive (line 5). At the beginning of each time slot t, the system has to schedule data for transmission (line 6). V_t represents the amount of video data to transmit at time t. $V_t = B_t \Delta T$ when considering the most expensive interface, otherwise, $V_t = (\beta + 1) B_t \Delta T$. $\beta \in \mathbb{R}^+$ is a parameter that controls the optimism of the scheduler. When $\beta = 0$, the scheduler tries to upload the content at the minimum rate which guarantees to complete the upload within the deadline. When $\beta > 0$, the system is more optimistic, and the scheduler tries to utilize any excess bandwidth, so as to deliver more data, and stay ahead of schedule. This increases the chance of completing the upload before the deadline, even if the available bandwidth drops below the expected value in the future. In other words, the parameter β allows pushing extra data into the interface buffer to efficiently use the excess bandwidth of cheap interfaces. By forcing $\beta = 0$ for the most expensive interface[5], we avoid any extra load on that interface, in order to minimize its use and the overall cost of upload (line 8). The amount of data scheduled on interface i at time t is the minimum between V_t and the estimated expected rate $\widehat{r}_{ti} \Delta T$ (line 12). line 13 computes any leftover of video still to schedule on more expensive interfaces.

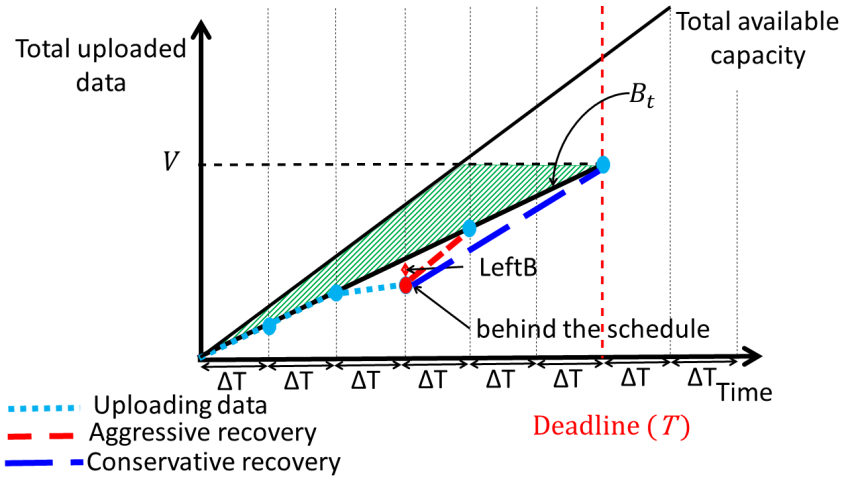


Fig. 6.6 Scheduling process over time and recovery strategies

The data in the buffer is transmitted over all interfaces (line 14). At the end of each slot, the algorithm updates the amount of remaining data that must still be transmitted (line 16), and updates the transmission rate (line 18), and the expected rate (line 19-24).

[5]When costs change over time, the algorithm adapts β consequently.

If the aggregate transmission rate is smaller than expected, data is accumulated in $LeftB$. When this happens, the system has to recover by increasing the amount of data to schedule for transmission B_{t+1}. If the *aggressive* recovery policy is selected, the scheduler tries to recover in the immediately following slot (line 29). If instead the *conservative* recovery policy is selected, the scheduler spreads the left-over data over all the remaining slots until the deadline, which leads to a higher average transmission rate in remaining slots (line 31).

Fig. 6.6 illustrates the evolution of the upload process over time. At each ΔT, the algorithm schedules the amount of data to be uploaded B_t. At the end of the third time slot, the actual amount of transmitted data is lower than expected, due to a drop in bandwidth. The system reacts by updating the expected rate for future slots, and trying to either recover in the immediately upcoming slot (aggressive policy, red line), or in the remaining slots before the deadline (conservative policy, blue line).

6.5 Trace-driven Simulation

6.5.1 Simulation Setup

We first describe the simulation setup that we used to run experiments, aiming at the performance evaluation of different schedulers, at the identification of suitable parameter values, and at the exploration of trade-offs between performance and complexity. We consider a scenario where 2 videos must be uploaded from a vehicle equipped with one node with 3 different network interfaces, each one using a different technology: 3G, 4G, and WiFi. As we already mentioned, we base our experiments on the traces presented in Section 6.3. Since public transport vehicles repeatedly follow a fixed path, we loop the traces as many times as necessary to reach the deadline. To allow for some randomness (inherent in wireless bandwidth availability, due to varying network conditions), we select a random combination of traces for each simulation run, and we choose a random starting point for each trace.

The video upload deadline T is chosen in the order of a few minutes. The time slot duration is the only parameter for which domain knowledge can offer a compelling choice: ΔT must be coherent with the time scale of changes in bandwidth at the different interfaces. Using large values for ΔT decreases the ability of the scheduler to adapt to bandwidth changes in a timely manner, whereas having very

small values results in unnecessary oscillations and lower bandwidth utilization. The traces collected for WiFi, 3G, and 4G show bandwidth changes on a scale of seconds. For this reason, we choose for ΔT a value equal to 1 second. [6]

The cost associated with each interface is an input to the scheduler, and can be chosen according to the end user constraints and the specific context of the application; it can be derived from either tariffs, or energy consumption, or data quota, or system load, or a combination of those. In our context, the node is onboard a bus, and always plugged; therefore, we assume that energy consumption is not crucial. Considering that the proposed adaptive scheduler is a lightweight application, we do not consider system load to define the cost. We thus use the monetary cost associated to the cost per bit on each interface (hence with each technology). Considering the end user fees for data transmission over different technologies (and including flat as well as variable fees). Arbitrarily, we assume the cost assigned with each interface to be 2, 4, and 8 $(Mb)^{-1}$, respectively, for WiFi, 3G, and 4G. Note that we assume that costs are the same for all videos, but the scheduler can cope with costs that are different for each video. This feature can be exploited, together with the selection of different deadlines, when videos have different urgency. In our previous work [139] we also considered a larger scenario with 10 interfaces and 5 videos, but we omit it here for the sake of brevity.

Videos have size equal to 62.5 MB ($V = 125$ MB in total), corresponding to about 5 minutes of 1080p video. The simulation time (which corresponds to the deadline T) varies in the range of [100,1000] seconds. We repeat each experiment 100 times with a random combination of traces as input, and we measure the average and the confidence interval for the following two metrics: i) time to complete the upload; and ii) cost of the upload.

6.5.2 Perfect Knowledge Centralized Scheduler Results

We start by considering the case of perfect knowledge of available bandwidth in future slots, and by comparing the performance of the MCFP to the one of the greedy heuristics. We use the *IBM ILOG CPLEX Optimization Studio 12.6.0.0* [172] Solver Engine to find the optimal solution of the MCFP formulation, while the greedy heuristics are implemented in Python. Experiments were run on the high performance

[6] We explored ΔT values in $\{1, 2, 5, 10\}$ s. Results are omitted for the sake of brevity.

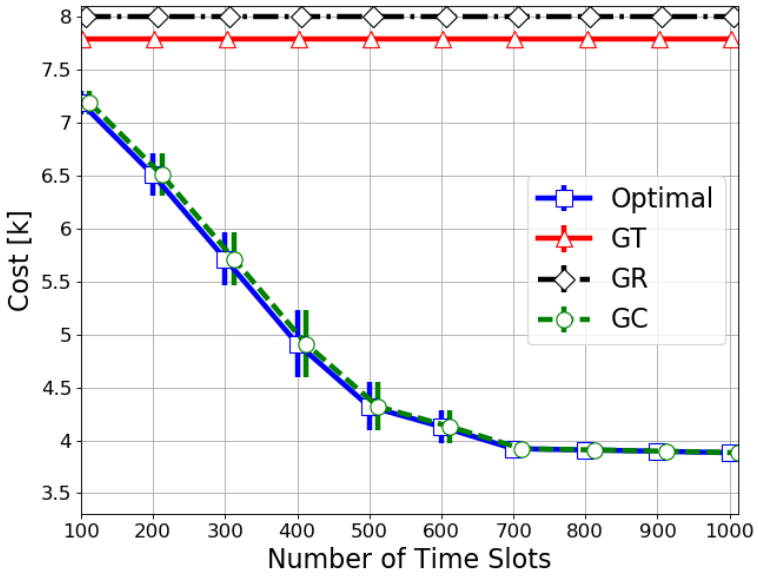


Fig. 6.7 Cost for perfect knowledge centralized results

computing cluster *hpc@polito*.[7] Intuition suggests that, for growing values of the deadline, performance should improve, since schedulers have more opportunities to trade cost against delay. In addition, we can expect that schedulers force the upload completion times to be close to the deadline, waiting for any cheap slot appearing toward the end of the available time interval (given the perfect knowledge of available bandwidth in future slots).

Fig. 6.7 reports results for the average upload cost (together with confidence intervals) versus the upload deadline (expressed in number of time slots) for the greedy heuristics and the optimal solution. As expected, the GR algorithm incurs the highest upload cost, followed by GT. Both algorithms are insensitive to the deadline value, since they do not consider cost. On the contrary, the GC algorithm and the optimal solution provide cost values which decrease for growing deadline, thus meeting our expectation. Quite interesting is the fact that the GC algorithm provides results that are marginally higher than those of the optimal solution.

Fig. 6.8 shows the average total upload time (and its confidence interval) as a function of duration of upload (in terms of the number of time slots) for different heuristics and optimal solution. As previously anticipated the optimal solution

[7] http://www.hpc.polito.it

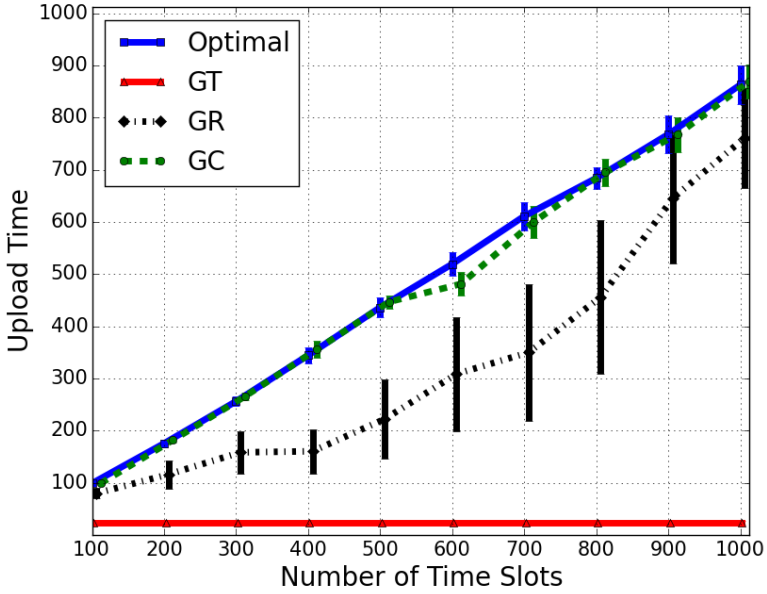


Fig. 6.8 Upload time for perfect knowledge centralized results

produces upload times very close to the deadline. The GC and GR also leverage the freedom to choose on multiple slots when the deadline is large. On the contrary, the GT algorithm yields very short upload completion times, as expected.

The two main conclusions that we can draw from this first set of results are: i) it is possible to reduce cost by effectively utilizing available bandwidth, despite substantial variations in available wireless bandwidth over time; ii) GR performs worse than GT, since the latter provides lower cost and lower completion times; iii) GC achieves practically the same cost and completion times as the optimal solution.

6.5.3 Dynamic Heuristic Scheduler Results

We now consider the realistic case where no a priori knowledge of available bandwidth is available. For the scheduler performance analysis, we implemented a custom simulator using Python. The simulator models the upload of K ($K = 2$ in our results) videos from a mobile vehicle equipped with WiFi, 3G, and 4G interfaces (one each in our results). Traces are used to emulate the actual available bandwidth at any time slot.

Dynamic Heuristic Scheduler Parameters

For starters, we investigate the scheduler parameter setting, to better understand the effect of parameter values on performance. The two scheduler parameters are α and β. The parameter α gets values in $[0,1]$ and drives the EWMA estimation of available bandwidth in future slots. The parameter β controls the amount of extra data (in addition to the estimated available bandwidth) pushed in all interface buffers, except the most expensive one, as described in Section 6.4.4. β takes values in $[0,30]$.

For some combinations of parameters, the upload of the video may not terminate within the specified deadline due to the improper scheduling strategy. We thus use these metrics to evaluate the performance of the scheduler: i) the total cost of the upload, ii) the probability of completing the video upload within the deadline, and iii) the amount of time after the deadline to complete the upload. Our objective is to achieve a very high completion probability with a very low cost.

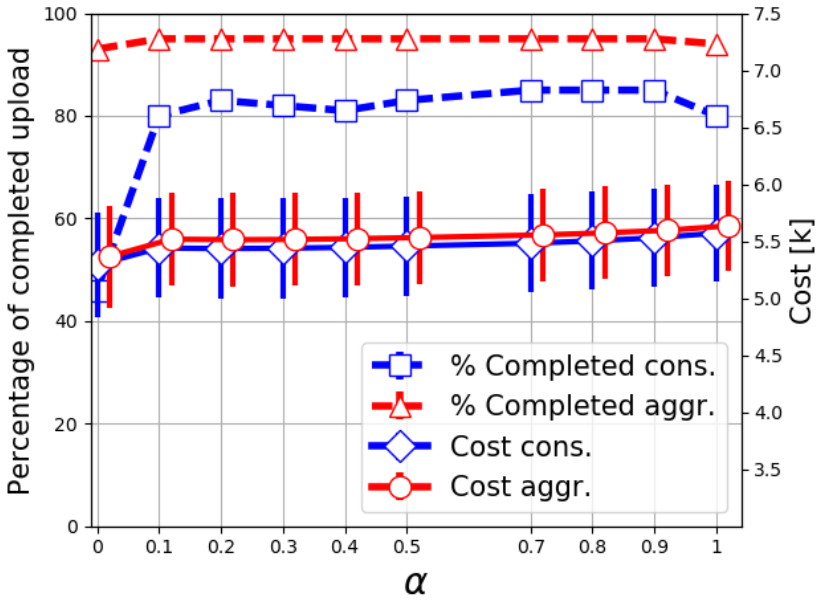


Fig. 6.9 Percentage of completed uploads and cost versus α with $T = 300$ s and $\beta = 1$

Parameter Setting

We first investigate the impact of α, which determines the timescale of the rate estimation for each interface. Fig. 6.9 presents the percentage of completed uploads (dotted lines - left y-axis) and their total cost (solid lines - right y-axis) as a function of α for different recovery approaches. Notice how performance is rather insensitive to the value of α. Recalling that α drives the EWMA estimation of the $\widehat{r}_{ti}$, we can conclude that it does not have considerable impact on the proposed adaptive scheduler. This holds true also for all considered β not reported here for the sake of brevity. That is, even a very coarse estimation of the link available bandwidth is sufficient to achieve our goals. In the following, we fix $\alpha = 0.1$.

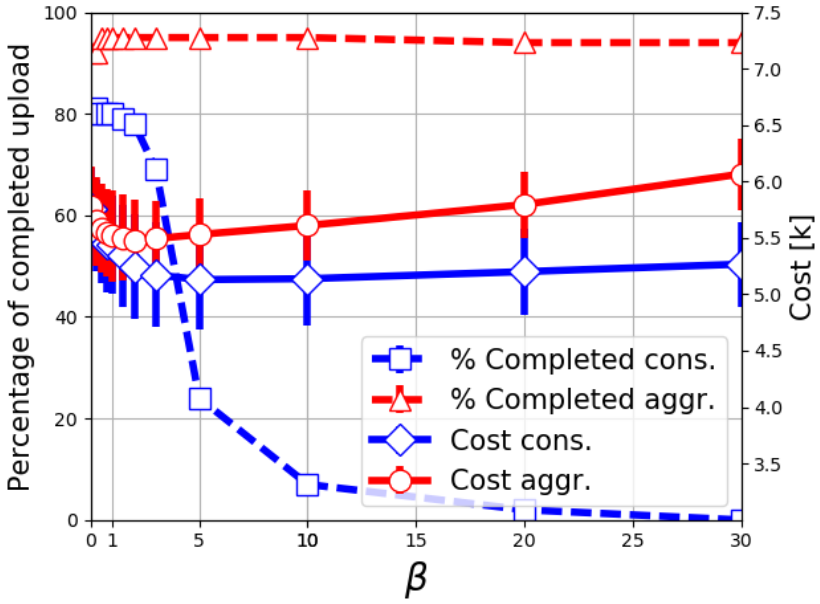
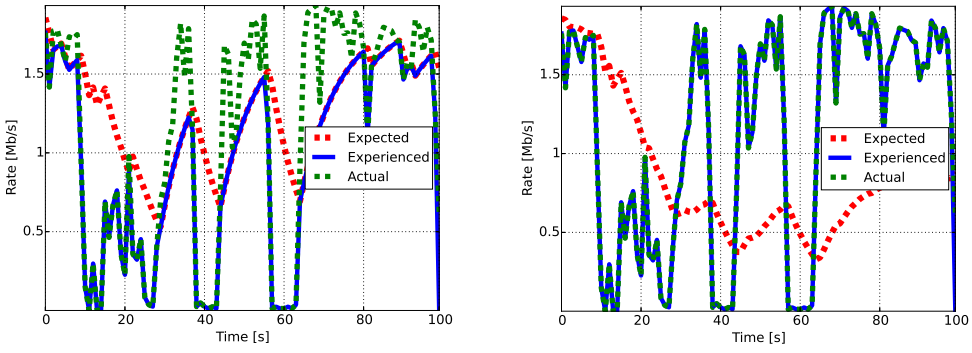


Fig. 6.10 Percentage of completed uploads and cost versus β with $T = 300$ s and $\alpha = 0.1$. Trying to push more data than expected has positive benefits on the aggressive algorithm, but dramatic effects on the conservative algorithm

The choice of β is less intuitive. To illustrate this, Fig. 6.10 shows the percentage of completed uploads (dotted lines - left y-axis) and their total cost (solid lines - right y-axis) as a function of β. We clearly see that β has a significant impact on performance. Fig. 6.10 indicates that the aggressive recovery algorithm (red curves) is less sensitive to β (lines are almost flat), which means that aggressively recovering is more important than optimistically spreading extra data across remaining time slots. The blue curves show that the conservative approach does not work properly

with large values of β. The rate of delivery required to catch up may never become available, thus missing the deadline. Indeed, increasing β makes the system try to optimistically send more data through cheaper interfaces with the aim to reduce the overall upload cost. However, due to insufficient bandwidth across the remaining slots, the gap between the expected and the actual pace of progress of the upload ($LeftB$) keeps increasing. Therefore, the conservative recovery strategy is unable to catch up and meet the deadline. Notice how the chance to meet the deadline suddenly decreases for increasing β, with most uploads failing for $\beta > 5$. In the following, we fix $\beta = 1$.

To further illustrate the effect of β on the performance of the scheduling algorithm, Fig. 6.11 shows the evolution of the upload rate over the 3G interface versus time during an experiment, with $\beta = 0$ and $\beta = 5$, respectively. The green (dashed line) is the available bandwidth, the red (dashed line) is the EWMA of available rate, and the blue (solid line) is the experienced rate. The closer the blue curve is to the green curve, the more the system is able to exploit the available bandwidth. Fig. 6.11 clearly proves that the choice $\beta = 0$ lets a large fractions of the actual available bandwidth on 3G go unused. This forces the scheduler to use the expensive 4G interface. Setting $\beta > 0$ makes the system more optimistic, and prone to send more data than the current bandwidth estimation would allow. This increases the utilization of the 3G interface and hence reduces the load on the (expensive) 4G interface.



(a) 3G with $\beta = 0$ (b) 3G with $\beta = 5$

Fig. 6.11 The aggressive scheduler operation with $\beta = 0$ (upper), $\beta = 5$ (lower), and $\alpha = 0.1$

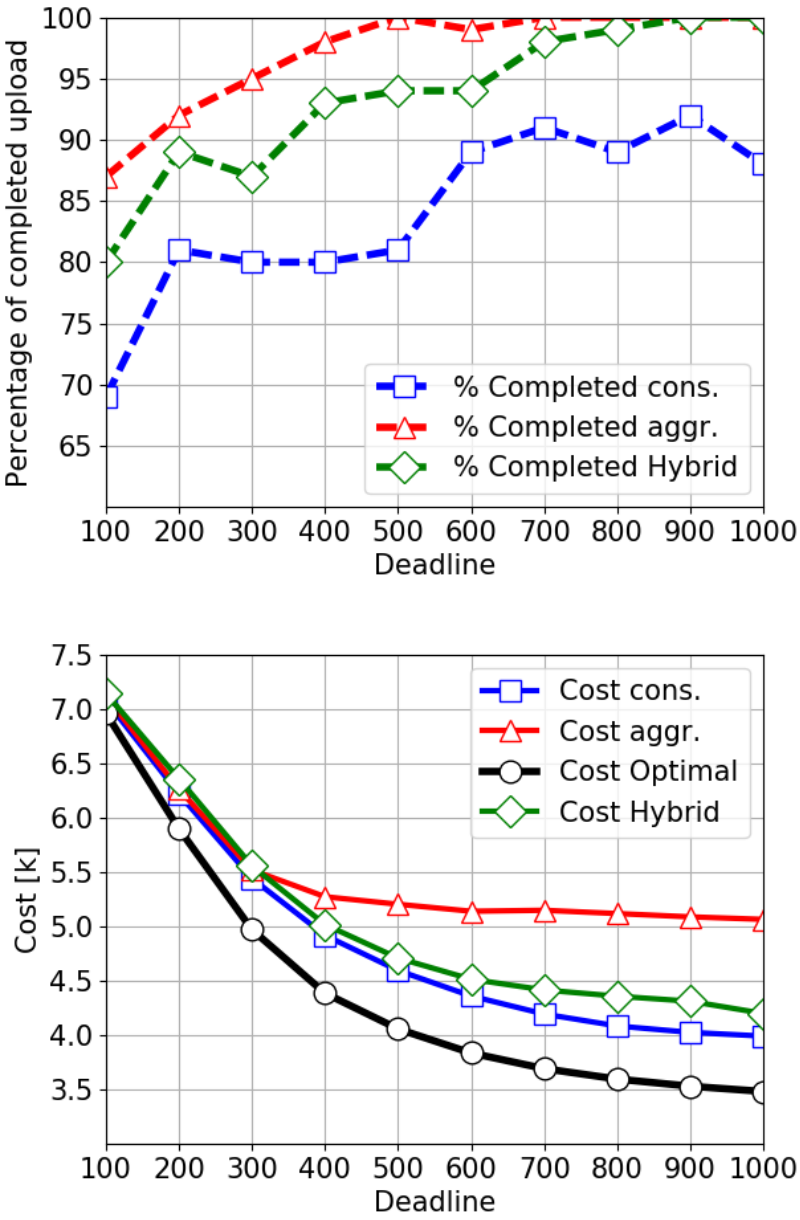


Fig. 6.12 Percentage of completed uploads (top) and cost (bottom) versus deadline T, with $\beta = 1$ and $\alpha = 0.1$

Impact of the Deadline

We now look at the influence of the deadline on performance. Fig. 6.12 shows the percentage of completed uploads (top plot) and their total cost (bottom plot), for values of $T \in [100, 1000]\,s$, with $\beta = 1$, and $\alpha = 0.1$. As we can expect, longer

deadlines imply higher percentages of completed uploads, and lower costs. Looking in more detail at the percentage of completed uploads, we see that the aggressive version of the algorithm (red curve) consistently outperforms the conservative version (blue curve). As previously observed, the latter suffers in scenarios where bandwidth becomes scarce when approaching the deadline.

To appreciate the performance of the adaptive scheduler with respect to the total upload cost, we compare it against the straightforward GT heuristic, that uploads all data as fast as possible, and (as we already saw) provides overall costs very close to the optimal solution. On average, the GT scheduler completes the upload in 33 s, with a cost of 7.5 k units. Adaptive schedulers reduce the cost to about 4 k units with conservative recovery and to about 5 k units with aggressive recovery, with savings of about 46%, and 33%, respectively. In general, the distance of the cost curves from the curve of the perfect knowledge case is quite small for very short deadlines, and remains within about 20-25% for longer deadlines. The former effect is due to the limited choice that short deadlines leave to the scheduler. The latter effect is a clear indicator of the good performance of our adaptive scheduling algorithm.

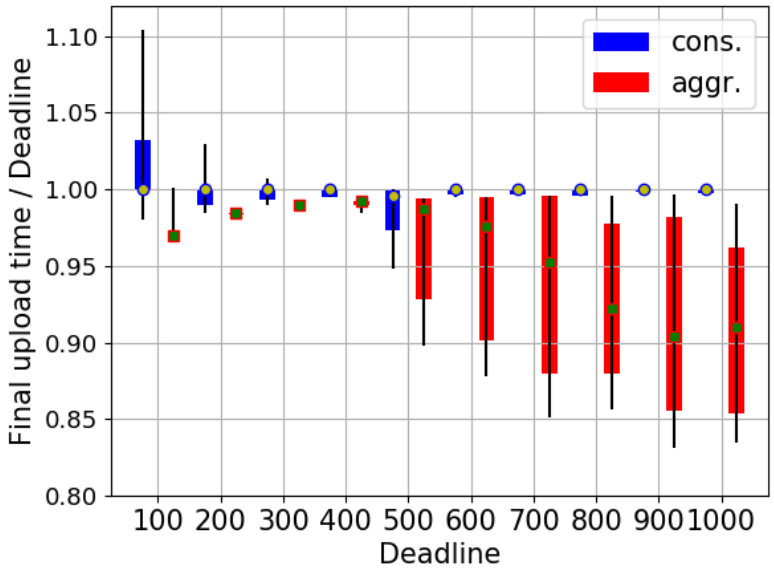


Fig. 6.13 Distribution of final upload time over deadline versus deadline, with $\beta = 1$ and $\alpha = 0.1$

Fig. 6.12 shows that the conservative recovery algorithm yields lower costs, but higher percentage of missed deadlines, with respect to the aggressive recovery

algorithm. It is interesting to see by how much the deadline is missed by the conservative recovery algorithm, i.e., how many more time slots the scheduler needs, to complete the video upload. Fig. 6.13 reports percentiles of the upload completion time, normalized to the deadline. Box-and-whiskers plots show the 90th, 80th, 20th and 10th percentiles of the upload time, for conservative (blue) and aggressive (red) recovery policies, respectively. Average values are represented by dots. Values larger than 1 show the fraction of extra time slots needed to complete the video upload. We can observe that about 10% more time slots in the case of the conservative recovery policy would allow almost 90% of successful schedulings, even in the case of tight deadlines.

While Fig. 6.13 considers all simulated schedulings, Fig. 6.14 consider only those that fail to meet the deadline, and reports the number of time slots needed to complete the upload, again with box-and-whiskers plots. Results show that for deadlines longer than 5 minutes (300 time slots), about 90% of the schedulings complete within a delay of half a minute (30 slots).

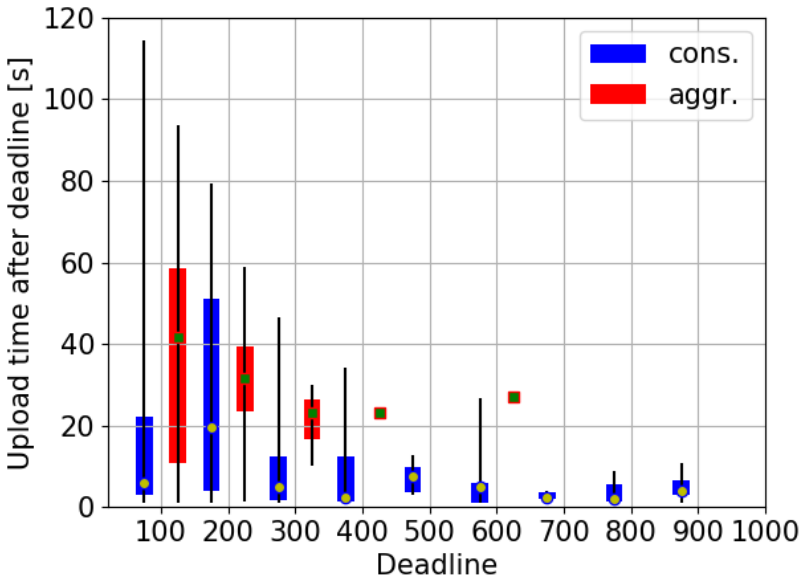


Fig. 6.14 Distribution of final upload time over deadline versus deadline, with $\beta = 1$ and $\alpha = 0.1$ for schedulings that do not meet the deadline

Hybrid Algorithm

The fact that i) the conservative recovery algorithm achieves lower costs, and ii) the aggressive algorithm guarantees higher chances to meet the deadline, motivates a hybrid approach that tries to combine the strengths of the two approaches. In particular, we suggest to use the conservative approach in the initial part of the scheduling, while switching to the aggressive approach when getting closer to the deadline. Intuitively, at the beginning of the scheduling, the hybrid algorithm tries to decrease the cost by using a conservative recovery. The choice of $\beta = 1$ allows to push extra data into cheaper interfaces, avoiding the most expensive one. To prevent accumulating too much unsent data, and taking the risk of missing the deadline, the hybrid algorithm switches to the aggressive recovery policy when approaching the deadline.

We evaluate the performance of this hybrid approach, by using the conservative recovery for the first 90% of slots, and then switching to the aggressive recovery algorithm in the remaining 10% of slots. The green line in Fig. 6.12 shows that this hybrid algorithm achieves very high completion probability with very competitive cost (only about 15% higher than the oracle). This shows that the hybrid approach can properly leverage the tradeoff between cost minimization and and upload time, even under unpredictable variations in available bandwidth.

Impact of Available Bandwidth Variability

The proposed adaptive scheduler exploits the long-term average of the available bandwidth, together with an EWMA, to predict the available bandwidth in future slots. The effectiveness of this strategy clearly depends on the variability of the available bandwidth. It is thus important to investigate what is the acceptable variance range for performance to be good. To this end, we add variance to the available bandwidth measured in our traces, by letting each sampled available bandwidth value $r_{t,i}$ become a random variable with uniform distribution in the range $[r_{t,i}(1-X), r_{t,i}(1+X)]$, with parameter X in [0,1]. The resulting variance added to the sampled available bandwidth value $r_{t,i}$ is:

$$\sigma^2_{r_{t,i}} = \frac{X^2 r^2_{t,i}}{3} \tag{6.15}$$

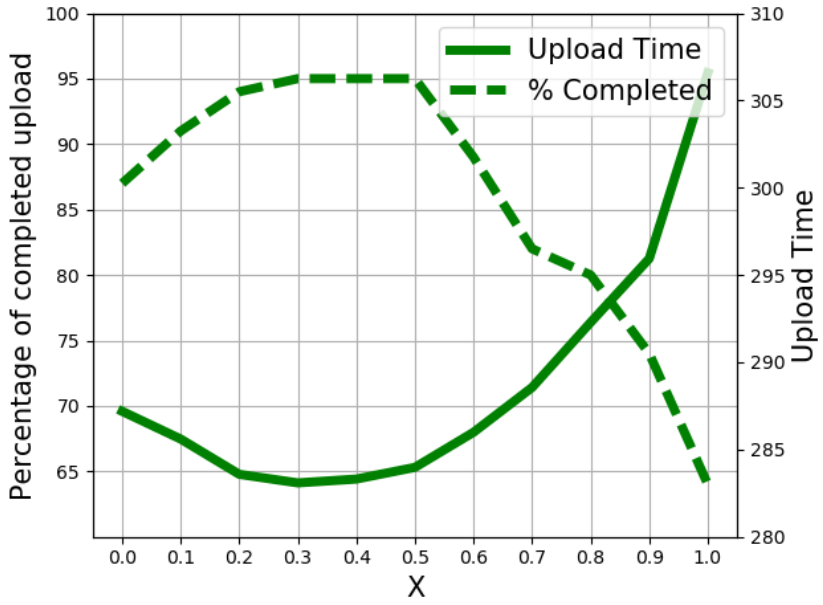


Fig. 6.15 Effect of variation on interface available bandwidth, with deadline = 300, $\beta = 1$, and $\alpha = 0.1$

Fig. 6.15 shows the percentage of completed uploads (dashed line - left y-axis) and their upload time (solid line - right y-axis) for the hybrid algorithm, versus X, with deadline 300 s, $\beta = 1$, and $\alpha = 0.1$. The reported values are the averages of 100 repetitions including the ones that failed to meet the deadline. Results indicate that the scheduler behaves well up to $X = 0.5$. A variance increases, the percentage of completed upload within the deadline can drop by 30%. Interestingly, moderate variance helps the algorithm to both reduce the cost and meet the deadline. Because, $\beta > 0$ lets the algorithm to exploit extra bandwidth on cheap interfaces. Conversely, when the randomness increase over 50%, the unpredictability of the system reduce the performance. Notice that the average completion time remains close to the deadline.

6.6 Experimental Evaluation

In this section we describe the design, implementation, and test of our adaptive scheduler in multihomed nodes deployed on vehicles in the framework of the MONROE project.

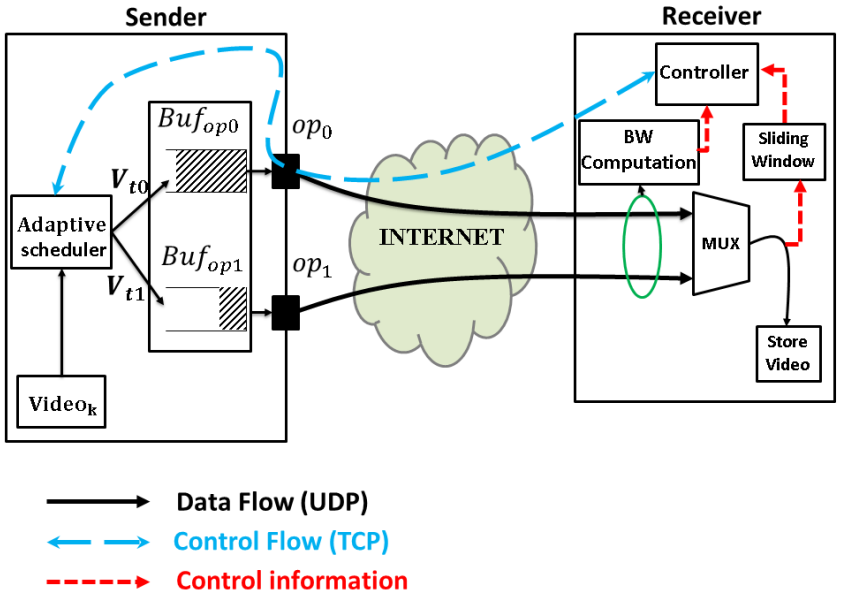


Fig. 6.16 Internal architecture of the adaptive scheduler implementation

6.6.1 Experimental Setup

We first discuss the experimental setup and the engineering choices adopted in the implementation of the adaptive scheduler.

Protocol Issues

At the experiment start, the mobile node (the video sender) registers into the central office node (the video receiver). For this, the sender initiates a TCP connection that is used for signaling, and waits for a request from the central office node. In case of disconnection, the sender re-registers itself. This choice simplifies the connection handling, e.g., avoiding NAT traversal issues, commonly encountered in today's MBB networks [133, 174]. For the video upload, the choice of the transport protocol can affect both the performance of the scheduler, and the actual implementation complexity.

In the experiment scenario, TCP would need some application mechanisms to manage connections (one for each interface), which may be complex in our mobile scenario, especially for the WiFi interface, which has to reopen a connection each

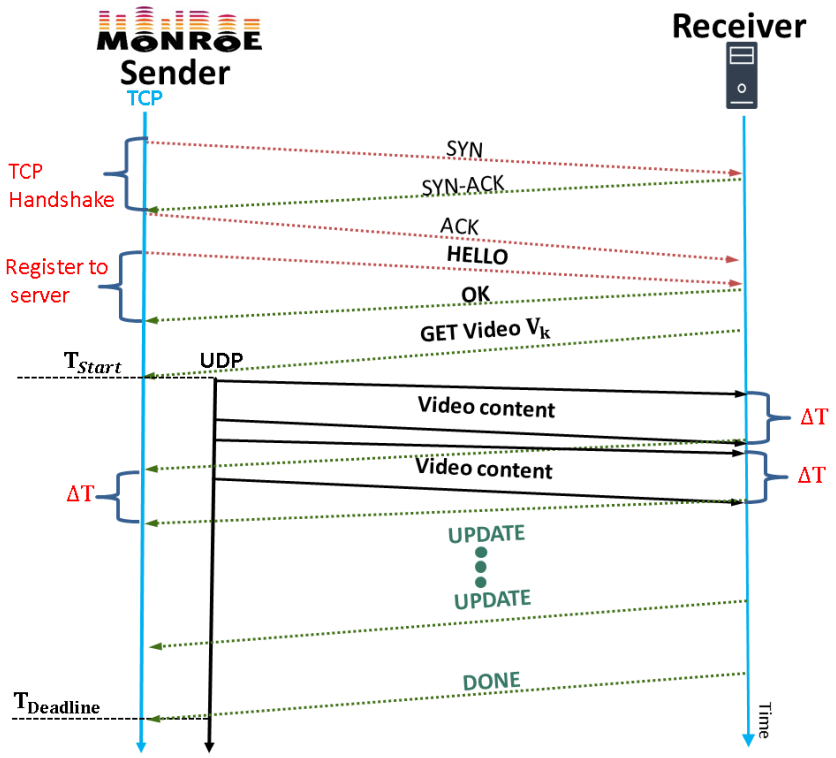
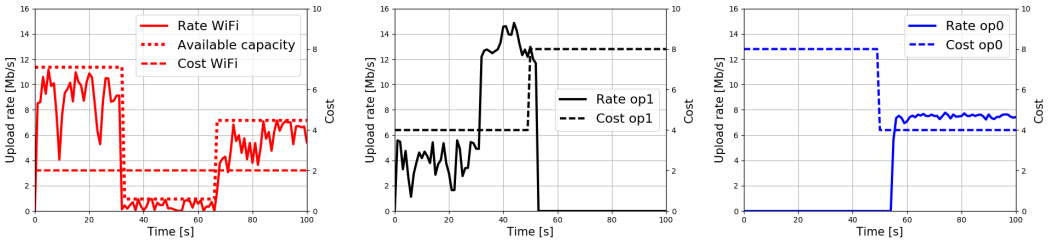


Fig. 6.17 Experiment design

time it joins a new hotspot. TCP also complicates the sender buffer management, since data can be stored in the sender TCP buffer with no knowledge of how much has been actually delivered to the receiver side. This is an issue during transition phases, e.g., when switching the assigned packet from one interface to another one, since an assigned packet to an interface is pushed into its buffer and is not accessible. UDP on the contrary offers a datagram and unreliable service, with accurate delimitation of messages. This gives us the freedom to push data into the interfaces with tighter control, letting the scheduling algorithm decide the sending rate for each interface. However, UDP offers no guarantee on the delivery of messages. For this, we need to add an Automatic Repeat reQuest (ARQ) protocol [175].

In our prototype, we opt for UDP as transport protocol, and we implement a selective repeat ARQ protocol, with acknowledgements that report information on the received messages using a bitmap of 10,000 elements. Acknowledgements are generated every second, and sent on the TCP signaling channel. Acknowledgements also carry information to accurately compute the actual rate at the receiver side for each interface. The receiver calculated the data rate for each interface every



(a) WiFi (b) op1 (c) op0

Fig. 6.18 Adaptive scheduler experiment on a stationary MONROE node

second by using the information obtained from packet header. The header contains information about the node, video, interfaces, and data offset.

Experiment Architecture

Fig. 6.16 illustrates the internal architecture of the adaptive video upload scheduler implemented in the MONROE platform. The left side represents the sender (the public transport vehicle which uploads the video). The sender is equipped with multiple MBB interfaces. The right side represents the receiver (the security center) that requests the specific content with a given deadline. A TCP connection is used as a control channel to register the node, request the content, and send update and control messages. Update messages contain the upload rate computed at the receiver, as well as acknowledgements carrying the bitmap of the received segments to implement the ARQ protocol. The control connections consume about 1% of the video upload bandwidth.

Fig. 6.16 illustrates the control flow (TCP) and data flow (UDP) using blue and black arrows, respectively. The adaptive scheduler module runs the video upload scheduling algorithm and pushes data into interface buffers. The MUX module stores packets for each video on disk, while the sliding window module keeps track of all the packets that were already received. Since the adaptive scheduler module at the sender side needs the information about the experienced interface rate, every ΔT the controller module sends update messages and a bitmap of the missing packets (implementing a selective repeat ARQ protocol).

Fig. 6.17 shows the communication workflow in our experiment. All the connections are initiated from the sender. At the beginning, the sender starts a TCP

connection (on a reliable interface - MPTCP could be used here) as control channel. After the TCP three-way handshake, the sender registers itself with an HELLO message. The HELLO message contains information about the available interfaces and videos. If the server recognizes the client, it replies with an OK message. If the TCP connection fails at any time, the client starts a new TCP connection.

The server can request a specific video portion by issuing a GET message with a specific chunk ID. Then, the sender starts sending UDP messages on selected interfaces to upload data according to the hybrid adaptive video upload scheduler. The receiver replies with UPDATE messages to report the last time slot data rate, and the bitmap of received messages. If for any reason the UPDATE message does not arrive at the end of each time slot, the sender assumes all packets of the previous slot are delivered successfully and keeps sending messages. With the next UPDATE message, the bitmap will arrive, possibly missing messages will be retransmitted, and rate adjusted. Messages are 1,000 Byte long. With selective acknowledgements carrying the status of the last 10,000 messages, we have an equivalent window of 10 MB. Since ACKs are sent every second, this lets us reach a data rate equal to 80 Mb/s. [8]

6.6.2 Experimental Results

We started our experimental analysis by using stationary nodes with three interfaces: 2 MBB interfaces, named op0 and op1, and 1 Ethernet wired connection, which was used to emulate a WiFi connection [9] by using the Linux traffic control tool[10] to limit the bandwidth and impose random (1%) packet loss. We assumed that the interface costs change over time, to be able to illustrate and evaluate the behavior of the video upload scheduling algorithm.

Fig. 6.18 shows all interface upload rates (solid lines, left y-axis) and costs (dashed lines, right y-axis) versus time in seconds. The dotted line in Fig. 6.18a shows the available bandwidth of the WiFi interface (emulated on the Ethernet connection). In the interval [35,65] seconds we simulate a period of lack of coverage to see if the scheduler is going to use more expensive interfaces. Fig. 6.18b shows

[8] 8*1000*10000=80Mb/s

[9] At the time of the experiment campaign, the MONROE nodes were not yet incorporating a WiFi interface

[10] http://lartc.org/manpages/tc.txt

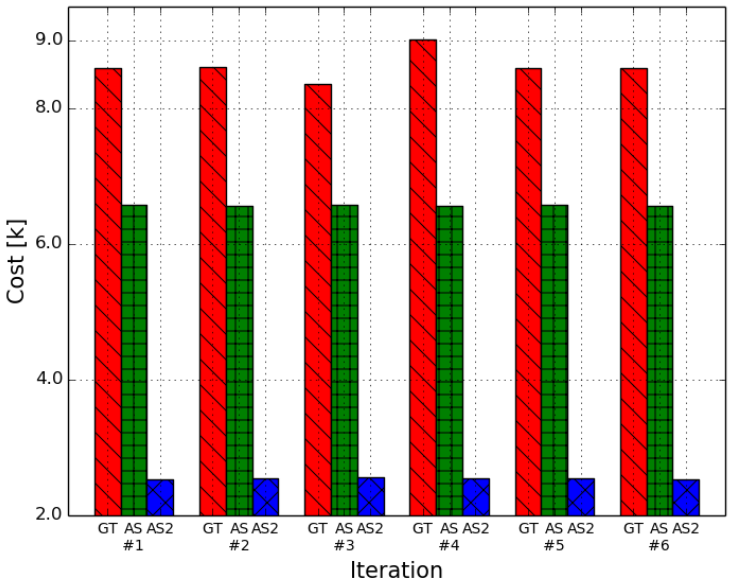


Fig. 6.19 Cost comparison of GT-100 (GT), AS-100 (AS), and AS-200 (AS2) in MONROE node traveling on public transport vehicle in 6 iterations

that op1 is selected to support the transmission of data that cannot make it through the WiFi interface. The cost of the interfaces op0 and op1 changes after 50 seconds, and the adaptive scheduler switches to the now cheaper interface to upload data. This proves that the video upload scheduler operates as expected.

We next present results obtained by running experiments on mobile MONROE nodes equipped with two 4G MBB interfaces and one WiFi interface. We compare the results obtained with the hybrid video upload scheduler against those obtained with the GT heuristic, which is the only heuristic that can schedule the video content with no a priori knowledge of the interface available bandwidth.

We compare the total cost of the GT (GT-100) algorithm, and of the adaptive scheduler (AS-100) with deadline of 100 s, and of the adaptive scheduler (AS-200) with deadline of 200 s. In order to allow the reader to get a quantitative view of the scenarios experienced by the tests, consider that RTT, available bandwidth, and RSSI were measured in the ranges $[40,120]$ ms, $[0,80]$ Mb/s, and $[-30,-100]$ dBm, respectively. The tests for GT-100, AS-100, and AS-200 were run back to back with 20 seconds idle time (batch of experiment), repeating each experiment 6 times. Fig. 6.19 shows the total costs (y-axis) for the three schedulers in 6 batch of experiments (x-axis). We can see that AS-100 (AS) obtains solutions about 30%

cheaper than GT-100 (GT), while completing the upload within the deadline. As expected, AS-200 (AS2) uploads the video with 40% lower cost with respect to AS-100 (AS).

6.7 Conclusions and Outlook

In this chapter, we considered the practical problem of video surveillance in connected public transport vehicles, where security videos are stored onboard, and a central operator sometimes requests to watch portions of the videos. At these requests, the selected video must be uploaded within a given deadline, by using the wireless network interfaces available on the vehicle, considering interfaces have different associated costs. The video upload goal is to minimize the total cost of the upload while meeting the deadline.

To identify an effective solution to our problem, we explored several aspects. In order to obtain a benchmark, we first considered the case where an oracle has perfect knowledge about available bandwidth of wireless links; we formalized the corresponding optimization problem and proposed greedy heuristics. Second, we looked at the case where the distribution of the available bandwidth on wireless interfaces is known; we defined the corresponding stochastic optimization problem, then we found that its solution is computationally extremely costly. We thus explored a family of adaptive scheduling algorithms that require only a coarse knowledge of the available bandwidth on wireless interfaces. Results show that our adaptive scheduling approach can effectively leverage the fundamental tradeoff between the total video upload cost and the video delivery time, despite unknown short-term variations in throughput on wireless links. Finally, we implemented and tested our adaptive algorithm in the platform for wireless network experiments provided by the MONROE project.

We believe, there are particular aspects of the algorithm can be evaluate more deeply. As the first step, we plan to compare the performance of the proposed adaptive scheduler with MP-DASH and other open source video adaptive scheduler. Secondly, the parameters can be define by dynamic approach instead of in advance tuning. The cost also can be modeled for the general multihomed MBB devices.

Chapter 7

Conclusions

Mobile network becomes one of the main tools to access the Internet meanwhile the Internet architecture becomes more complex and hard to study. Traffic measurement plays a crucial role in analyzing and understanding the behavior of different players in such a sophisticated environment. Moreover, MBB networks bring some other difficulties by adding mobility, technology diversity, and etc., into an already dynamic system. There is a grave need for an open measurement platform, geographically distributed to collect large-scale measurements along with analyzing tools to shed light on the performance, users' QoE of the MBB networks, and evaluation of innovative applications.

In the following, we shortly summarize the most relevant findings and contributions for the subjects tackled.

First, we reported on our experience designing an open large-scale measurement platform for experimentation with commercial MBB networks (Chapter 2). MONROE is a completely open system, allowing authenticated users to deploy their own custom experiments and conduct their research in the wild. The platform is a crucial means to understand, validate, and ultimately improve how current operational MBB networks perform towards providing guidelines to the design of future 5G architectures. We described our experiences with the MONROE system implementation and detailed the hardware selection for the MONROE measurement node, its software ecosystem and the user access and scheduling solution. We emphasized the versatility of our proposed design, both for the overall platform and, more specifically, for the measurement nodes. In fact, the node software design is compatible with a number

of different hardware implementations, given that it can run on any Linux-compatible multihomed system. Our current hardware solution is the most fitting for the set of requirements and the predicted usage of MONROE, which we evaluated based on our discussions and interaction with the platform's users.

In context of the performance measurement, we focus on network layer performance, users' QoE on web, and understanding of roaming in Europe. Specifically in Chapter 3, we discussed our experience in running "speedtest-like" measurements to evaluate the download speed offered by actual 3G/4G networks. Our experiments were permitted by the availability of the MONROE platform. Despite our test simplicity, download speed measurements in MBB networks are much more complex than in wired networks, because of many factors which clutter the picture. The analysis of the results indicated how complex it is to draw conclusions, even from an extended and sophisticated measurement campaign. As a result, the key conclusion of our work is that benchmarks for the performance assessment of MBB networks are badly needed, in order to avoid simplistic, superficial, wrong, or even biased studies, which are difficult to prove false. Defining benchmarks that can provide reliable results is not easy, and requires preliminary investigation and experience.

Chapter 4 presented a cross-European study of web performance on commercial mobile carriers using the MONROE system. The novelty of the study stands in the sheer volume of data we were able to collect from MONROE nodes operating under similar conditions in 11 different MBB networks. Our results and further analysis brought to light the complexity of the cellular networks, where the randomness of the wireless access channel coupled with the often unknown operator configurations makes monitoring performance very challenging. We found that the overall web performance is similar across different countries and operators, with only slight variations. In aggregate per target websites, our measurements showed that the performance improvements H2 promised still remain to be experienced. Furthermore, we found that web performance is mainly dependent on the characteristics and performance of the target web page. Thus, for websites where we conjecture that the server-side implementation of H2 is more mature (Youtube) we observed superior performance from the end-user perspective.

Chapter 5 illustrated our observation of roaming in Europe. Different network configuration options can affect performance of various applications for the end user in roaming. In practice, there are three possible solutions (i.e., HR, LBO, and

IHBO), we find that HR is exploited by the 16 operators in our experiments. This comes with performance penalties on the roaming user, who experiences increased delay and appears to the public Internet as being connected in the home country. This has further implications in the selection of CDN server when roaming abroad, because the mobile user will access a server in the home network rather than one close to their location. However, in the same time, the roaming user is still able to access (in majority of cases) the geo-restricted services from the home country in its native language. We put these results in perspective while also trying to speculate on the commercial implications of the "Roam like Home" initiative.

In chapter 6, we opted for the practical problem of video surveillance in connected public transport vehicles, where security videos are stored onboard, and a central operator sometimes requests to watch portions of the videos. At these requests, the selected video must be uploaded within a given deadline, by using the wireless network interfaces available on the vehicle, considering interfaces have different associated costs. The video upload goal is to minimize the total cost of the upload while meeting the deadline. To identify an effective solution to our problem, we explored several aspects. In order to obtain a benchmark, we first considered the case where an oracle has perfect knowledge about available bandwidth of wireless links; we formalized the corresponding optimization problem and proposed greedy heuristics. Second, we looked at the case where the distribution of the available bandwidth on wireless interfaces is known; we defined the corresponding stochastic optimization problem, then we found that its solution is extremely costly computation wise. We thus explored a family of adaptive scheduling algorithms that require only a coarse knowledge of the available bandwidth on wireless interfaces. Results showed that our adaptive scheduling approach can effectively leverage the fundamental tradeoff between the total video upload cost and the video delivery time, despite unknown short-term variations in throughput on wireless links. Finally, we implemented and tested our adaptive algorithm in the platform for wireless network experiments provided by the MONROE project. We believe, there are particular aspects of the algorithm that can be evaluated in more detail. Firstly, parameters can be defined by a dynamic approach instead of in advance tuning. Secondly, the cost can be modeled for the general multihomed MBB scenario.

Appendix A

Statistical Distance Measures

In this thesis we selected a specific Statistical Distance Measure (SDM) that we used as $F(p,q)$, namely the Jensen-Shannon divergence (JS_{div}). The purpose of this section is thus to (i) contrast the broad set of SDMs from both a theoretic viewpoint, as well as making punctual examples to narrow down SDMs selection; (ii) show that, due to functional relationships between SDMs, it is possible to express the same methodology with multiple equivalent metrics, among which the Jensen-Shannon divergence (JS_{div}); and (iii) assess robustness of the $F(p,q)$ estimation as function of the p,q population size and binning strategy employed.

A.1 SDM Comparison

In this thesis, we do not aim at proposing a novel SDM. We instead prefer to collect a set of well-known and established SDM available in literature, analyze their features and choose the most suitable one for our use-case. Gibbs et al. [176] compare a variety of SDMs, shedding light on their properties and on the relationships among them. Without aiming at completeness, we report in Tab. A.1 a list of 9 representative SDMs considered in [176], plus the SDM proposed in [177]. Specifically, for each SDM the table reports its name, abbreviated notation, definition, co-domain and three relevant properties: (i) *Metric*, the SDM is a function defining a metric distance between each pair of elements in a set; (ii) *Bounded*, the SDM co-domain is finite; and (iii) *Symmetric*, the SDM is invariant to which of the two distributions is considered the reference, i.e., $F(p,q) = F(q,p)$.

Name	Abbrv	Formula	Image	Properties		
				Metric	Bounded	Symmetric
Jensen-Shannon	JS	$JS_{div}(p,q) = \sum_i \left\{ \frac{1}{2} p_i \ln\left(\frac{p_i}{\frac{1}{2}p_i + \frac{1}{2}q_i}\right) + \frac{1}{2} q_i \ln\left(\frac{q_i}{\frac{1}{2}q_i + \frac{1}{2}p_i}\right) \right\}$	$[0, ln(2)]$		✓	✓
Kullback-Leibler	KL	$KL_{div}(p,q) = \sum_i p_i log(\frac{p_i}{q_i})$	$[0,\infty)$			
Chi Square	χ^2	$\chi^2_{dis}(p,q) = \sum_i \frac{(p_i - q_i)^2}{q_i}$	$[0,\infty)$			
Separation	S	$S_{dis}(p,q) = max_i\left(1 - \frac{p_i}{q_i}\right)$	$[0,1]$		✓	
Total variation	TV	$TV_{dis}(p,q) = \frac{1}{2}\sum_i \lvert p_i - q_i \rvert$	$[0,1]$		✓	✓
Hellinger	H	$H_{dis}(p,q) = \left[\sum_i (\sqrt{p_i} - \sqrt{q_i})^2\right]^{\frac{1}{2}}$	$[0,\sqrt{2}]$		✓	✓
Kolmogorov	K	$K_{met}(P,Q) = \sup_x \lvert P(x) - Q(x) \rvert$	$[0,1]$	✓	✓	
Wasserstein	W	$W_{met}(P,Q) = \int_{-\infty}^{\infty} \lvert P(x) - Q(x) \rvert dx$	$[0,1]$	✓	✓	
Discrepancy	D	$D_{met}(P,Q) = \sup_{\text{all closed balls B}} \lvert p(B) - q(B) \rvert$	$[0, diam\Omega]$	✓	✓	
DCF09 [177]	L	$L_{div}(p,q) = \frac{1}{2}\left(\frac{KL_{div}(p,q)}{E_p} + \frac{KL_{div}(q,p)}{E_q}\right)$	$[0,\infty)$			✓

Table A.1 Statistical Distance Measures. In the above formulas, p and q denote two empirical distributions on the measurable space Ω, with p_i and q_i being their samples, and P and Q their cumulative distribution functions. Note that in L, E_x is the entropy of empirical distribution x (we preferred to use E instead of the common H notation to avoid conflicts with H – Hellinger).

From Tab. A.1 it is easy to see a rather heterogeneous picture. Most SDMs are divergence measures, with the exclusion of Kolmogorov (K), Wasserstein (W) and Discrepancy (D), which are metrics. With the exception of Kullback-Leibler (KL) and Chi-Square (χ^2), all other SDMs have a bounded co-domain. Finally, only Jensen-Shannon (JS), Total Variation (TV) and Hellinger (H) are symmetric. At last, we explicitly consider the metric proposed in [177] which is symmetric, but not bounded and not a metric. None of the SDMs exhibits all three properties. As we shall see later, these properties play an important role in the SDM selection.

In terms of provenance and use, JS and KL are information theoretic measures. Loosely speaking, KL expresses the amount of information that is required to encode q knowing p, while JS expresses the average amount of information carried by q which is not in p. χ^2, H and K are often used for statistical tests. [177] has been proposed to specifically tackle anomaly detection in network measurements context.

We finally broaden the investigation by considering the SDMs introduced earlier in Tab. A.1, with the aim for both highlighting the relationships among them, as well as illustrating the behavior of each SDM in simple scenarios early considered for the JS_{div}. In principle, any of the SDMs in Tab. A.1 can fit the purpose of our

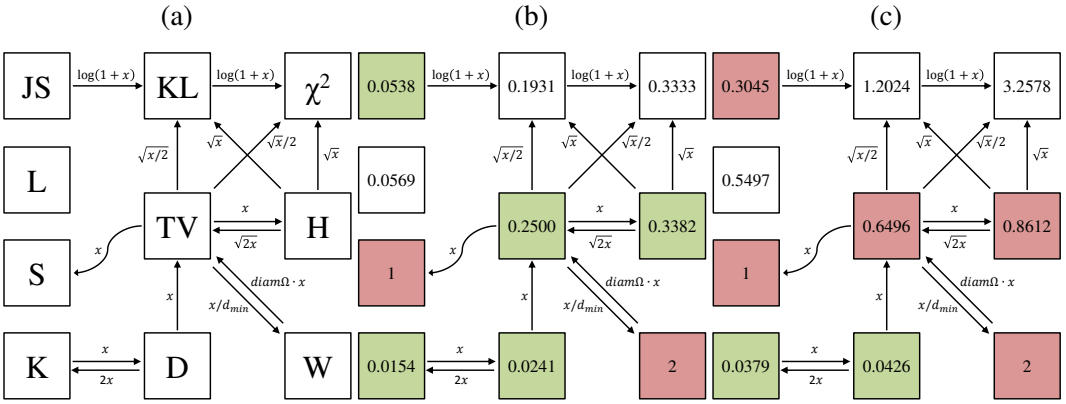


Fig. A.1 (a) Overview of distance measures: a directed arrow from A to B annotated by a function h(x) means that $A \leq h(B)$. The symbol $diam\Omega$ denotes instead the diameter of the probability space Ω, and for Ω finite, $d_{min} = \inf_{x,y \in \Omega} d(x,y)$. (b) reports the values assumed by the considered metrics when we match a negative exponential distribution with λ_0=1 against a second one with λ_1=2. Similarly, (c) reports the values when λ_1=8.

framework, so we illustrate here some relevant criteria to narrow down the SDM selection to a small set of equivalent functions.

Following the same approach presented in Fig. 1 of [176], we arrange SDMs as a matrix of blocks in Fig. A.1a. To show the relationship among them, Fig. A.1a provides bounds between metrics. Given any two distance metrics A and B, a directed arrow from A to B annotated by a function $h(x)$ means that $A \leq h(B)$. For instance, take the Discrepancy (D) and Kolmogorov (K) metrics: the x and $2x$ arrows simply encode the $K_{dis} \leq D_{dis} \leq 2K_{dis}$ inequality.

With the exception of L, it can be seen that, since each considered SDM is directly related to at least another one, their dependency graph consists of a single connected component, i.e., it is possible to find bounds among different SDMs in practice.

Fig. A.1b and Fig. A.1c show illustrative examples of these SDMs by comparing synthetic probability distributions. The aim is to visualize how the earlier illustrated SDM properties decline from practical viewpoint, and how these properties can be leveraged to narrow down SDM selection. For the sake of space, we focus here on the case of negative exponential distributions.

Several considerations hold contrasting Fig. A.1b and Fig. A.1c. First, Separation (S) and Wasserstein (W) saturate to the upper bound already with λ_1=2, so that their

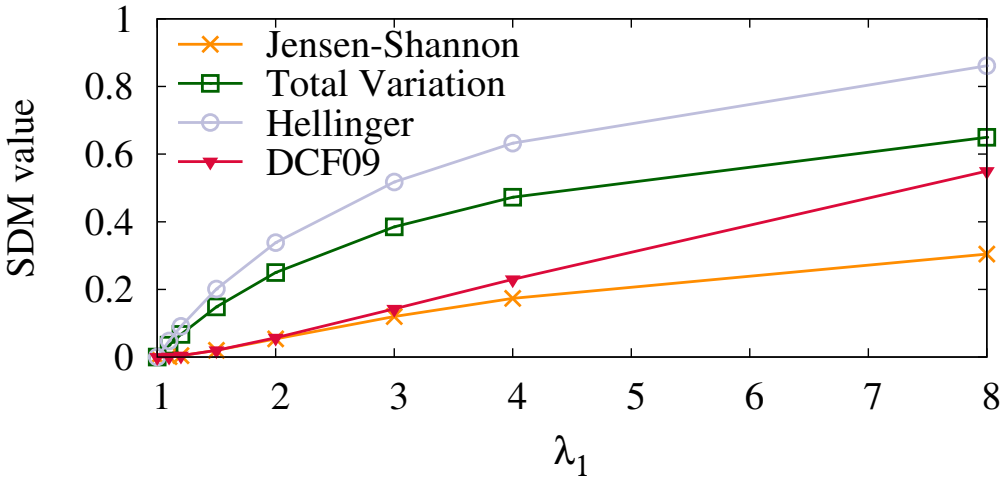


Fig. A.2 Values specific SDMs assume when matching a negative exponential distribution with λ_0=1 against a second one with parameter λ_1 varying in $[1,8]$.

fast varying dynamic is not able to express the greater difference among λ_0=1 and λ_1=8.

Second, Kolmogorov (K) and Discrepancy (D) report a very low value both in case of λ_1=2 and λ_1=8, with a slow dynamic that is opposite to the previous case. For all S, W, K and D, it is clear that the impact of Q^- and Q^+ threshold selection in $Q(F(p,q))$ becomes of paramount importance; additionally, no Q^- and Q^+ selection would allow to express differences for SDMs such as S and W.

Third, L, KL and χ^2 show a good sensitivity to the changes of the λ_1 parameter. However, they are all unbounded measures (which makes them not practical), and KL and χ^2 are not symmetric (which rules them out from our framework). Asymmetric metrics can be used when one wants to test a (suspect) population against a reference (well behaving). However the lack of symmetry makes it harder to use KL and χ^2 in the general case where one has no a priori expectation about a population.

Fourth, we observe that the remaining measures, namely Jensen-Shannon (JS), Total Variation (TV) and Hellinger (H) are all good candidates. Not incidentally, JS, TV and H are the sole SDMs in Fig. A.A.1a that are symmetric and bounded. In addition, they show to be consistently sensitive to different values of the λ_1 parameter, as better detailed in Fig. A.2. Any of JS, TV and H are equivalent to our purpose. For the sake of completeness, we include DCF09 (L) since it has been previously used in the field of network anomaly detection. The same consideration

holds, despite this SDM being not bounded. For practical purposes, it is pointless to perform exhaustive analytics with each of these metrics, which we show to be equivalent for our purpose. To avoid bringing redundant information and cluttering pictures and tables, we restrict our attention to JS_{div} as reference $F(p,q)$ measure.

A.2 JS_{div} Sensitivity Analysis

We now assess the SDM robustness to factors that may affect the EPDF estimation, as these may induce artificial errors leading to wrong conclusions. Indeed, the whole framework rely on the ability to compute a statistically relevant distance measure $F(p,q)$ between two population samples (represented by their EPDFs p and q). This distance measure $F(p,q)$ is then compared to two empirical thresholds Q^- and Q^+ to discriminate between cases having a practically negligible, practically noticeable or practically relevant significance. Of course, this *practical significance* holds only provided that $F(p,q)$ is also *statistically significant*, as otherwise differences between the population samples that p and q may be actually artifacts tied to a number of random fluctuations. Otherwise stated, the relevance of the framework is conditioned to the statistical significance of the computed metrics, as otherwise it would be possible to raise alarms that are however not statistically significant.

To avoid the above problem, we need not consider the potential source of errors that can indeed affect SDMs, of which the most prominent are: (i) the binning strategy used to compute the samples of p and q distributions; (ii) the imbalance in the population size of p and q; and (iii) the finitude of p and q populations.

A.2.1 Binning Strategy

Let us first start from the impact of the binning strategy. Taking JS_{div} as an example, we assess the operating conditions of the framework that ensure proper evaluation of the EPDFs. We expect the binning adopted in estimating the EPDF to play a role for continuous metrics with domain in $\mathbb{R}$: intuitively, coarse bins smooth down differences (JS_{div} decreases, approaching 0 in the limit case where all samples fall in the same single bin). Fine grained bins, in contrary, exacerbate differences (JS_{div} increases and approaches $\ln(2)$ for rational bins of vanishing size, each of which contains a single or few samples).

It is thus important to assess the settings of the uniform binning strategy, i.e., the support and bin size (or equivalently, number of bins). As done previously, we follow an engineering and experimental approach. We consider p and q as negative exponential distributions, with $\lambda_0 = 1,\ \lambda_1 \in \{2,4\}$. Given the previous Q^-, Q^+ thresholds, we expect $q = NegExp(x,2)$ to fall in the intermediate state, while $q = NegExp(x,4)$ to be significantly different from p. To avoid small population noise, we use finite sequences of 10^6 samples for each distributions. We then extract the empirical distributions from the two dataset by considering a number of bins which varies from 2 to 10^6. We limit the support in the $[0,100)$, thus $\Delta b \in [0.001, 50]$. We then compute the JS_{div} to compare p and q. For each value of the bin, we repeat 100 runs.

Fig. A.3a show results, where the x-axis reports the number of bins, and the y-axis the corresponding JS_{div} value. Note the logarithmic scales. When the number of bins is smaller than 50, a underfitting phenomenon emerges, so that the JS_{div} artificially drops to small values. Similarly, when the number of bins grows larger than 5,000, an overfitting phenomenon is visible, so that the JS_{div} artificially increases. We see that the JS_{div} is consistent for number of bins in the 50-5,000 range, where the EPDFs are correctly estimated. The inset details the relative error that occurs to JS_{div} with respect to the value obtained when using 50 bins, i.e., the reference. The relative error is below 19%. It follows that quantization oddities are controllable, provided a large number of samples is available, and that the support of the distribution is limited.

In general, it is good practice to select a binning strategy that is tied to the physics of the metric: for example, use an unitary bin size for measurements that takes integer values (e.g., the Number of Hops, of SYN messages, etc.), or relate the bin size to the unit of scale of interest (e.g., a 1 ms accuracy for RTT and time-related metrics, or consider bins of 10 kbps when dealing with throughput). This calls for ingenuity and suggest the involvement of domain expertise.

In presence of heavy-tailed distributions, the choice of logarithmic binning strategies, or of mixed linear-logarithmic ones as suggested in [177], could be considered. By using logarithmic binning one would alleviate the problem of vanishing bins with few samples (which typically would occur in the tail of the distribution), and limit the number of bins. However, this comes at a cost. Notice indeed that engineering questions would arise: How many bins should be used, and how to properly set

the switching threshold from linear to logarithmic binning? All these choices have, in our opinion, to be driven by domain knowledge and should be tailored to the application domain.

A.3 Population Size

Clearly, a specular question is in place: for a given bin size choice, which is the impact of the number of samples on the estimation of the EPDF? Intuitively, while any finite sequence deviates from quantiles of the theoretic distribution, small population samples tend to exhibit larger deviations.

Taking two finite realizations of the same process, we estimate the empirical EPDFs p and q and compute the JS_{div}. To avoid binning errors, we consider real-valued distributions (i.e., Gaussian, negative exponential) and an integer-valued distribution (Geometric). We then estimate the two (nominally identical) EPDFs using a number of samples that varies from 10 to 10^5 samples. We compute the JS_{div} (which we expect to be close to 0) considering 1,000 bins.

Fig. A.3b show results. Irrespectively of the distribution, JS_{div} is strongly affected by the population size (linear slope in log-log plot). As expected, an excessively small population inflates the JS_{div} value. Specifically, having less than 1,000 (100) samples in the population causes the JS_{div} to exceed the warning threshold for noticeable (significant) differences for all the distributions. It is thus recommended to employ the JS_{div} on population larger than 1,000 samples, assumption verified in our dataset.

However, it is important to mention that artifacts caused by a limited population size may have an impact in case the methodology is used in real-time (e.g., on short time window) scenarios, or to compare the same population over different temporal samples. This possibly mandates a minimum duration of the observation period, especially in off-peak times, so to reach a minimum level of observation samples.

Finally, population imbalance is worth discussing, as it may introduce yet another bias. Yet, we experimentally observe that, as long as the smallest population is statistically significant, then no noticeable bias appear – which is intuitive since EPDFs renormalize the contribution of each population.

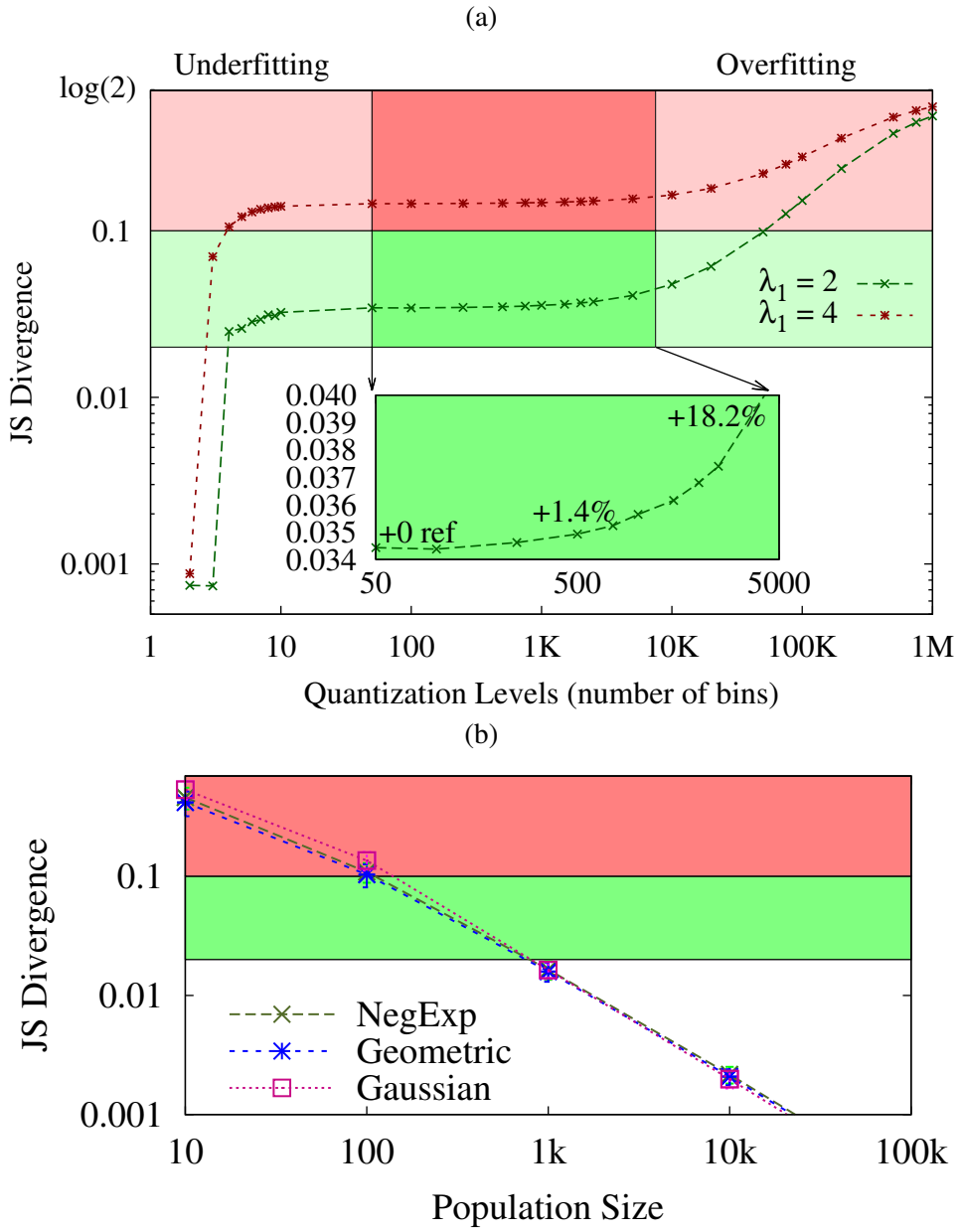


Fig. A.3 Sensitivity analysis of Jensen-Shannon divergence for: (a) varying number of bins, (b) varying population size for two finite realization of the same process.

Appendix B

About Author

In this part, I (**Ali Safari Khatouni**) present a short summary of my research activities during my PhD and biography which might be a better to describe them separately from the thesis main thread.

B.1 Biography

I was born in Iran. I graduated my B.S. Degree in Computer Engineering (Software) from Urmia University, Iran in 2007. I received my M.Sc. Degree in Computer and Communication Networks Engineering from Politecnico di Torino, Italy in 2014. During my M.Sc. thesis, I visited the Bell Labs, Alcatel-Lucent in France.

I joined the Telecommunication Networks Group of Politecnico di Torino as PhD candidate in 2015 under supervision of Prof. Marco Mellia. I participated in the mPlane (an Intelligent Measurement Plane fo Future Network and Application Management) European project in the first year. Then, I've actively participated in the Monroe (Measuring Mobile Broadband Networks in Europe) H2020 European project. More detail information is available in my personal webpage.

B.1.1 Publications

I involved in several research topics, beside the main topic presented in this thesis. In the following, I present full list of my publications during my PhD.

1. **Submitted: Ali Safari Khatouni**, Marco Ajmone Marsan, Marco Mellia, Reza Rejaie, Deadline-Constrained Content Upload from Multihomed Devices: Formulations and Algorithms, Computer Networks (COMNET), 2018

2. Miguel Peon-Quiros, Vincenzo Mancuso, Vincenzo Comite, Andra Lutu, Ozgu Alay, Stefan Alfredsson, Jonas Karlsson, Anna Brunstrom, Marco Mellia, **Ali Safari Khatouni**, Thomas Hirsch, Results from running an experiment as a service platform for mobile networks, The 11th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH2017), Snowbird, Utah, USA, October 20, 2017

3. Ozgu Alay, Andra Lutu, Miguel Peon-Quir, Vincenzo Mancuso, Thomas Hirsch, Kristian Evensen, Audun Hansen, Stefan Alfredsson, Jonas Karlsson, Anna Brunstrom, **Ali Safari Khatouni**, Marco Mellia, Marco Ajmone Marsan, Experience: An Open Platform for Experimentation with Commercial Mobile Broadband Networks, MobiCom, the 23th Annual International Conference on Mobile Computing and Networking, Snowbird, Utah, USA, October 16-20, 2017

4. **Ali Safari Khatouni**, Martino Trevisan, Leonardo Regano, Alessio Viticchie, Privacy Issues of ISPs in the Modern Web, IEEE IEMCON 2017, Vancouver, BC, Canada, October 3-5, 2017

5. **Ali Safari Khatouni**, Marco Mellia, Marco Ajmone Marsan, Stefan Alfredsson, Jonas Karlsson, Anna Brunstrom, Ozgu Alay, Andra Lutu, Cise Midoglu, Vincenzo Mancuso, Speedtest-like Measurements in 3G/4G Networks: the MONROE Experience, 29th International Teletraffic Congress - ITC'17, Genoa, September 4-8, 2017

6. **Ali Safari Khatouni**, Marco Ajmone Marsan, Marco Mellia, Reza Rejaie, Adaptive Schedulers for Deadline-Constrained Content Upload from Mobile Multihomed Vehicles, The 23rd IEEE International Symposium on Local and Metropolitan Area Networks, Osaka, Japan, June 12-14, 2017

7. **Ali Safari Khatouni**, Marco Mellia, Luca Venturini, Massimo Gallo, Diego Perino, Performance Comparison and Optimization of ICN Prototypes, 2016 IEEE Global Communications Conference: Workshops: Information Centric Networking Solutions for Real World Applications, Washington, DC USA, December 8, 2016

8. Ozgu Alay, Andra Lutu, Rafael Garcia, Miguel Peon Quiros, Vincenzo Mancuso, Thomas Hirsch, Tobias Dely, Jonas Werme, Kristian Evensen, Audun Fosselie Hansen, Stefan Alfredsson, Jonas Karlsson, Anna Brunstrom, **Ali Safari Khatouni**, Marco Mellia, Marco Ajmone Marsan, Roberto Monno, Hakon Lonsethagen, Demo: MONROE, a distributed platform to measure and assess mobile broadband networks, ACM WiNTECH 2016, New York City, USA, October 3, 2016

9. Ozgu Alay, Andra Lutu, Rafael Garcia, Miguel Peon Quiros, Vincenzo Mancuso, Thomas Hirsch, Tobias Dely, Jonas Werme, Kristian Evensen, Audun Fosselie Hansen, Stefan Alfredsson, Jonas Karlsson, Anna Brunstrom, **Ali Safari Khatouni**, Marco Mellia, Marco Ajmone Marsan, Roberto Monno, Hakon Lonsethagen, Measuring and Assessing Mobile Broadband Networks with MONROE (Best Demonstration), 17th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Coimbra, Portugal, June 21-24, 2016

10. Enrico Bocchi, **Ali Safari Khatouni**, Stefano Traverso, Alessandro Finamore, Maurizio Munafò, Marco Mellia, Dario Rossi, Statistical Network Monitoring: Methodology and Application to Carrier-Grade NAT, Computer Networks "Special issue on Machine learning, data mining and Big Data frameworks for network monitoring and troubleshooting", June, 2016

11. **Ali Safari Khatouni**, Marco Ajmone Marsan, Marco Mellia, Delay Tolerant Video Upload from Public Vehicles, Smart Cities and Urban Computing (SmartCity 2016), San Francisco, CA, April 11, 2016

12. **Ali Safari Khatouni**, Marco Ajmone Marsan, Marco Mellia, Video Upload from Public Transport Vehicles using Multihomed Systems, 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS): Student Activities, San Francisco, CA, April 10, 2016

13. Enrico Bocchi, **Ali Safari Khatouni**, Stefano Traverso, Alessandro Finamore, Valeria Di Gennaro, Marco Mellia, Maurizio Munafò, Dario Rossi, Impact of Carrier-Grade NAT on Web Browsing, 6th International Workshop on TRaffic Analysis and Characterization (TRAC 2015), Dubrovnik, Croatia, August, 2015

B.1.2 Awards

I have been awarded best paper and travel grants during my Ph.D. I list my awards, as follows:

• Best Paper Award at TRaffic Analysis and Characterization (TRAC), 2015.

• Best Demonstration Award at 17th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM), 2016.

• TMA 2015 Travel Grant.

• IMC 2015 Travel Grant.

• RIPE RACI 2018 Travel Grant.

References

[1] Mustafa Ergen. *Mobile Broadband - Including WiMAX and LTE*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[2] Ericsson. Ericsson Mobility Report 2016. White paper, 2016.

[3] Planetlab. https://www.planet-lab.org/.

[4] RIPE Atlas. https://atlas.ripe.net/.

[5] CAIDA. Archipelago (ark) measurement infrastructure, http://www.caida.org/projects/ark/.

[6] OOKLA. http://www.speedtest.net/.

[7] Christian Kreibich, Nicholas Weaver, Boris Nechaev, and Vern Paxson. Netalyzr: illuminating the edge network. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 246–259. ACM, 2010.

[8] Matthias Hirth, Tobias Hobfeld, Marco Mellia, Christian Schwartz, and Frank Lehrieder. Crowdsourced network measurements: Benefits and best practices. *Computer Networks*, 90:85–98, 2015.

[9] Mah-Rukh Fida, Andra Lutu, Mahesh Marina, and Ozgu Alay. ZipWeave: Towards Efficient and Reliable Measurement based Mobile Coverage Maps. May 2017.

[10] Transmission Control Protocol. RFC 793, September 1981.

[11] Jim Griner, John L. Border, Markku Kojo, Zach D. Shelby, and Gabriel Montenegro. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135, June 2001.

[12] European Commission: New Rules on Roaming Charges and Open Internet. https://ec.europa.eu/digital-single-market/en/news/new-rules-roaming-charges-and-open-internet. [Online; accessed 06-March-2018].

[13] V. Bajpai and J. Schönwälder. A survey on internet performance measurement platforms and related standardization efforts. *IEEE Communications Surveys Tutorials*, 17(3):1313–1341, thirdquarter 2015.

[14] Brian Trammell, Lianshu Zheng, Sofía Silva Berenguer, and Marcelo Bagnulo. Hybrid Measurement using IPPM Metrics. Internet-Draft draft-trammell-ippm-hybrid-ps-01, Internet Engineering Task Force, February 2014. Work in Progress.

[15] Kjell Brunnström, Sergio Ariel Beker, Katrien De Moor, Ann Dooms, Sebastian Egger, Marie-Neige Garcia, Tobias Hossfeld, Satu Jumisko-Pyykkö, Christian Keimel, Mohamed-Chaker Larabi, Bob Lawlor, Patrick Le Callet, Sebastian Möller, Fernando Pereira, Manuela Pereira, Andrew Perkis, Jesenka Pibernik, Antonio Pinheiro, Alexander Raake, Peter Reichl, Ulrich Reiter, Raimund Schatz, Peter Schelkens, Lea Skorin-Kapov, Dominik Strohmeier, Christian Timmerer, Martin Varela, Ina Wechsung, Junyong You, and Andrej Zgank. Qualinet White Paper on Definitions of Quality of Experience, March 2013. Qualinet White Paper on Definitions of Quality of Experience Output from the fifth Qualinet meeting, Novi Sad, March 12, 2013.

[16] A. Finamore, M. Mellia, M. Meo, M. M. Munafo, P. D. Torino, and D. Rossi. Experiences of internet traffic monitoring with tstat. *IEEE Network*, 25(3):8–14, May 2011.

[17] Martino Trevisan, Alessandro Finamore, Marco Mellia, Maurizio Munafo, and Dario Rossi. Traffic analysis with off-the-shelf hardware: Challenges and lessons learned. *Comm. Mag.*, 55(3):163–169, March 2017.

[18] A. Finamore, M. Mellia, M. Meo, and D. Rossi. Kiss: Stochastic packet inspection classifier for udp traffic. *IEEE/ACM Transactions on Networking*, 18(5):1505–1515, Oct 2010.

[19] M. Mellia, M. Meo, L. Muscariello, and D. Rossi. Passive identification and analysis of tcp anomalies. In *2006 IEEE International Conference on Communications*, volume 2, pages 723–728, June 2006.

[20] Ignacio N. Bermudez, Marco Mellia, Maurizio M. Munafo, Ram Keralapura, and Antonio Nucci. Dns to the rescue: Discerning content and services in a tangled web. In *Proceedings of the 2012 Internet Measurement Conference*, IMC '12, pages 413–426, New York, NY, USA, 2012. ACM.

[21] Gary S. Malkin. Traceroute Using an IP Option. RFC 1393, January 1993.

[22] Robert T. Braden. Requirements for Internet Hosts - Application and Support. RFC 1123, October 1989.

[23] Chung-Hsing Hsu and Ulrich Kremer. Iperf : A framework for automatic construction of performanceprediction. https://iperf.fr/, 2007.

[24] Karl Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, 1895.

[25] *Applied Nonparametric Regression*. Econometric Society Monographs. Cambridge University Press.

[26] networld2020. Service level awareness and open multi-service internetworking - principles and potentials of an evolved internet ecosystem, 2016.

[27] FCC. 2013 Measuring Broadband America February Report. Technical report, FCC's Office of Engineering and Technology and Consumer and Governmental Affairs Bureau, 2013.

[28] MONROE. Open source code. https://github.com/MONROE-PROJECT.

[29] DOCKER. https://www.docker.com/.

[30] MONROE. User access portal. https://www.monroe-system.eu.

[31] RaspberryPi. http://www.raspberrypi.org.

[32] Odroid. http://www.hardkernel.com.

[33] Beagleboard.org. http://beagleboard.org.

[34] PC Engines. APU2C4. https://www.pcengines.ch/apu2c4.htm.

[35] Yepkit. USB Switchable Hub YKUSH. https://www.yepkit.com/products/ykush.

[36] ZTE. USB-based CAT4 MF910 MiFi, product specification. http://www.ztemobiles.com.au/downloads/User_guides/MF910_Help1.0.pdf.

[37] Compex. WLE600VX: http://www.pcengines.ch/wle600vx.htm. http://www.pcengines.ch/wle600vx.htm.

[38] Sierra-Wireless. MC7455 miniPCI express (USB 3.0) modem. https://www.sierrawireless.com/products-and-solutions/embedded-solutions/products/mc7455/.

[39] ZeroMQ. http://zeromq.org/.

[40] JSON. http://www.json.org/.

[41] FED4FIRE. http://www.fed4fire.eu/.

[42] MONROE. Open-source experiments. https://github.com/MONROE-PROJECT/Experiments.

[43] Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Nicholas Weaver, and Vern Paxson. Beyond the radio: Illuminating the higher layers of mobile networks. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 375–387. ACM, 2015.

[44] Zhaoguang Wang, Zhiyun Qian, Qiang Xu, Zhuoqing Mao, and Ming Zhang. An untold story of middleboxes in cellular networks. In *Proc. of SIGCOMM*, 2011.

[45] Xing Xu, Yurong Jiang, Tobias Flach, Ethan Katz-Bassett, David Choffnes, and Ramesh Govindan. Investigating Transparent Web Proxies in Cellular Networks. In *Proc. of Passive and Active Measurement*, 2015.

[46] H2020 MAMI Project. Measurement and architecture for a middleboxed internet. https://mami-project.eu/.

[47] Tektronix. Reduce Drive Test Costs and Increase Effectiveness of 3G Network Optimization. Technical report, Tektronix Communications, 2009.

[48] Opensignal. https://opensignal.com.

[49] RTR-NetTest. https://www.netztest.at/en/.

[50] MobiPerf. http://www.mobiperf.com, 2014.

[51] Ashkan Nikravesh, David R. Choffnes, Ethan Katz-Bassett, Z. Morley Mao, and Matt Welsh. Mobile Network Performance from User Devices: A Longitudinal, Multidimensional Analysis. In *Procs. of PAM*, 2014.

[52] E. Halepovic, J. Pang, and O. Spatscheck. Can you GET me now?: Estimating the time-to-first-byte of HTTP transactions with passive measurements. In *Proc. of IMC*, 2012.

[53] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang. A first look at cellular network performance during crowded events. In *Proc. of SIGMETRICS*, 2013.

[54] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance. In *Proc. of SIGCOMM*, 2013.

[55] Zahir Koradia, Goutham Mannava, Aravindh Raman, Gaurav Aggarwal, Vinay Ribeiro, Aaditeshwar Seth, Sebastian Ardon, Anirban Mahanti, and Sipat Triukose. First Impressions on the State of Cellular Data Connectivity in India. In *Procs. of ACM DEV-4*, ACM DEV-4 '13, 2013.

[56] Džiugas Baltrūnas, Ahmed Elmokashfi, and Amund Kvalbein. Measuring the Reliability of Mobile Broadband Networks. In *Proc. of IMC*, 2014.

[57] Özgü Alay, Andra Lutu, Rafael García, Miguel Peón-Quirós, Vincenzo Mancuso, Thomas Hirsch, Tobias Dely, Jonas Werme, Kristian Evensen, Audun Hansen, et al. Measuring and assessing mobile broadband networks with monroe. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A*, pages 1–3. IEEE, 2016.

[58] Markus Laner, Philipp Svoboda, Peter Romirer-Maierhofer, Navid Nikaein, Fabio Ricciato, and Markus Rupp. A comparison between one-way delays in operating hspa and lte networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012 10th International Symposium on*, pages 286–292. IEEE, 2012.

[59] Alessio Botta and Antonio Pescapé. Monitoring and measuring wireless network performance in the presence of middleboxes. In *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*, pages 146–149. IEEE, 2011.

[60] Viktor Farkas, Balázs Héder, and Szabolcs Nováczki. A Split Connection TCP Proxy in LTE Networks. In *Information and Communication Technologies*, pages 263–274. Springer, 2012.

[61] N. Becker, A. Rizk, and M. Fidler. A measurement study on the application-level performance of lte. In *2014 IFIP Networking Conference*, pages 1–9, June 2014.

[62] Jie Hui, Kevin Lau, Ankur Jain, Andreas Terzis, and Jeff Smith. How YouTube Performance is Improved in T-Mobile Network. In *Proc. of Velocity*, 2014.

[63] F. Kaup, F. Michelinakis, N. Bui, J. Widmer, K. Wac, and D. Hausheer. Assessing the implications of cellular network performance on mobile content access. *IEEE Transactions on Network and Service Management*, 13(2):168–180, June 2016.

[64] Srikanth Sundaresan, Walter de Donato, Nick Feamster, Renata Teixeira, Sam Crawford, and Antonio Pescapè. Broadband internet performance: A view from the gateway. *SIGCOMM Comput. Commun. Rev.*, 41(4):134–145, August 2011.

[65] The TCP Maximum Segment Size and Related Topics. RFC 879, November 1983.

[66] David Borman, Robert T. Braden, Van Jacobson, and Richard Scheffenegger. TCP Extensions for High Performance. RFC 7323, September 2014.

[67] MAXMIND. https://www.maxmind.com/en/geoip2-isp-database.

[68] WHOIS. https://www.whois.net/.

[69] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[70] Marc C. Necker, Michael Scharf, and Andreas Weber. *Performance of Different Proxy Concepts in UMTS Networks*, pages 36–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[71] M. Ivanovich, P. W. Bickerdike, and J. C. Li. On tcp performance enhancing proxies in a wireless environment. *IEEE Communications Magazine*, 46(9):76–83, September 2008.

[72] HTTP Archive. http://mobile.httparchive.org/. [Online; accessed 11-June-2017].

[73] M. Belshe, R. Peon, and M. Thomson. Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540 (Proposed Standard), May 2015.

[74] Google Inc. Spdy: An experimental protocol for a faster web. https://www.chromium.org/spdy/spdy-whitepaper. [Online; accessed 11-June-2017].

[75] Steve Souders. Velocity and the bottom line. https://goo.gl/8wTE2e, Jul 2009. [Online; accessed 11-June-2017].

[76] Steve Souders. The performance of web applications – one-second wonders for winning or losing customers. https://goo.gl/ErNLcm, Nov 2008. [Online; accessed 11-June-2017].

[77] Eric Schurman (Bing) and Jake Brutlag (Google). Performance related changes and their user impact. https://goo.gl/hCr7ka, Jul 2009. [Online; accessed 11-June-2017].

[78] Ankit Singla, Balakrishnan Chandrasekaran, P. Brighten Godfrey, and Bruce Maggs. The internet at the speed of light. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, pages 1:1–1:7, New York, NY, USA, 2014. ACM.

[79] Latency is everywhere and it costs you sales - how to crush it. https://goo.gl/3PdqIP, Jul 2009. [Online; accessed 11-June-2017].

[80] Urs Hoelzle. The google gospel of speed. https://goo.gl/3LQyue, Jan 2012. [Online; accessed 11-June-2017].

[81] Philip Dixon. Shopzilla's site redo - you get what you measure. https://goo.gl/e9foNn, Jun 2006. [Online; accessed 11-June-2017].

[82] Ashkan Nikravesh, Hongyi Yao, Shichang Xu, David Choffnes, and Z Morley Mao. Mobilyzer: An open platform for controllable mobile network measurements. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 389–404. ACM, 2015.

[83] Joel Sommers and Paul Barford. Cell vs. wifi: on the performance of metro area mobile connections. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 301–314. ACM, 2012.

[84] John P. Rula, Vishnu Navda, Fabián Bustamante, Ranjita Bhagwan, and Saikat Guha. "No One-Size Fits All": Towards a principled approach for incentives in mobile crowdsourcing. In *Proc. of IMC*, 2014.

[85] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, and Jia Wang. Characterizing and Modeling Internet Traffic Dynamics of Cellular Devices. In *Proc. of SIGMETRICS*, 2011.

[86] S. Sen, J. Yoon, J. Hare, J. Ormont, and S.Banerjee. Can they hear me now?: A case for a client-assisted approach to monitoring wide-area wireless networks. In *Proc. of IMC*, 2011.

[87] R. Fielding et al. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999.

[88] Daniel Stenberg. Http2 explained. *SIGCOMM Computer Communication Review*, 44(3):120–128, July 2014.

[89] Michael Scharf and Sebastian Kiesel. Head-of-line blocking in tcp and sctp: Analysis and measurements, 2006.

[90] Torsten Zimmermann, Jan Rüth, Benedikt Wolters, and Oliver Hohlfeld. How HTTP/2 Pushes the Web: An Empirical Study of HTTP/2 Server Push. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, 2017.

[91] Matteo Varvello, Kyle Schomp, David Naylor, Jeremy Blackburn, Alessandro Finamore, and Konstantina Papagiannaki. Is the web HTTP/2 yet? In *Passive and Active Measurement - 17th International Conference, PAM*, pages 218–232, 2016.

[92] Matteo Varvello, Kyle Schomp, David Naylor, Jeremy Blackburn, Alessandro Finamore, and Kostantina Papagiannaki. To http/2, or not to http/2, that is the question. *arXiv preprint arXiv:1507.06562*, 2015.

[93] Kyriakos Zarifis, Mark Holland, Manish Jain, Ethan Katz-Bassett, and Ramesh Govindan. *Modeling HTTP/2 Speed from HTTP/1 Traces*, pages 233–247. Springer International Publishing, Cham, 2016.

[94] Enrico Bocchi, Luca De Cicco, Marco Mellia, and Dario Rossi. The web, the users, and the MOS: influence of HTTP/2 on user experience. In *Passive and Active Measurement - 18th International Conference, PAM*, pages 47–59, 2017.

[95] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. How speedy is spdy? In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 387–399, Seattle, WA, 2014. USENIX Association.

[96] Y. Liu, Y. Ma, X. Liu, and G. Huang. Can http/2 really help web performance on smartphones? In *2016 IEEE International Conference on Services Computing (SCC)*, pages 219–226, June 2016.

[97] Jeffrey Erman, Vijay Gopalakrishnan, Rittwik Jana, and Kadangode K Ramakrishnan. Towards a spdy'er mobile web? *IEEE/ACM Transactions on Networking*, 23(6):2010–2023, 2015.

[98] Yehia Elkhatib, Gareth Tyson, and Michael Welzl. Can spdy really make the web faster? In *Networking Conference, 2014 IFIP*, pages 1–9. IEEE, 2014.

[99] Greg White, Jean-François Mulé, and Dan Rice. Analysis of google spdy and tcp initcwnd. *Cable Television Laboratories, Inc*, 2012.

[100] H. de Saxcé, I. Oprescu, and Y. Chen. Is http/2 really faster than http/1.1? In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 293–299, April 2015.

[101] Utkarsh Goel, Moritz Steiner, Mike P Wittie, Martin Flack, and Stephen Ludin. Http/2 performance in cellular networks. In *ACM MobiCom*, 2016.

[102] Kyriakos Zarifis, Mark Holland, Manish Jain, Ethan Katz-Bassett, and Ramesh Govindan. Modeling http/2 speed from http/1 traces. In *International Conference on Passive and Active Network Measurement*, pages 233–247. Springer, 2016.

[103] Above-the-fold time (aft): Useful, but not yet a substitute for user-centric analysis. https://goo.gl/gWmj3r. [Online; accessed 11-June-2017].

[104] Steve Souders. Moving beyond window.onload(). https://goo.gl/rAeFNR. [Online; accessed 11-June-2017].

[105] Enrico Bocchi, Luca De Cicco, and Dario Rossi. Measuring the quality of experience of web users. *ACM SIGCOMM Computer Communication Review*, 46(4):8–13, 2016.

[106] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R Das. Improving user perceived page load times using gaze. In *NSDI*, pages 545–559, 2017.

[107] Speed index: Measuring page load time a different way. https://www.sitepoint.com/speed-index-measuring-page-load-time-different-way/. [Online; accessed 11-June-2017].

[108] Webpage test. https://www.webpagetest.org. [Online; accessed 11-June-2017].

[109] Utkarsh Goel, Moritz Steiner, Mike P Wittie, Martin Flack, and Stephen Ludin. Measuring what is not ours: A tale of 3rd party performance. In *ACM Passive and Active Measurements Conference (PAM)*, pages 142–155. Springer.

[110] Athula Balachandran, Vaneet Aggarwal, Emir Halepovic, Jeffrey Pang, Srinivasan Seshan, Shobha Venkataraman, and He Yan. Modeling web-quality of experience on cellular networks. In *Proceedings of the 20th annual international conference on Mobile computing and networking, MobiCom'14*, pages 213–224. ACM, 2014.

[111] Selenium – Web Browser Automation. http://www.seleniumhq.org, Online; accessed 11-June-2017.

[112] Xvfb manula page. https://goo.gl/EPBtpt. [Online; accessed 11-June-2017].

[113] HAR Export Trigger. http://www.softwareishard.com/blog/har-export-trigger/. [Online; accessed 19-June-2017].

[114] Alexa. The top sites on the web. http://www.alexa.com/topsites, 2017.

[115] GSM Association: IPX White Paper. https://www.gsma.com/iot/wp-content/uploads/2012/03/ipxwp12.pdf. [Online; accessed 06-March-2018].

[116] GSM Association: Guidelines for IPX Provider Networks. https://www.gsma.com/newsroom/wp-content/uploads//IR.34-v13.0-1.pdf. [Online; accessed 06-March-2018].

[117] Özgü Alay, Andra Lutu, Miguel Peón Quirós, Vincenzo Mancuso, Thomas Hirsch, Kristian Evensen, Audun Fosselie Hansen, Stefan Alfredsson, Jonas Karlsson, Anna Brunstrom, Ali Safari Khatouni, Marco Mellia, and Marco Ajmone Marsan. Experience: An Open Platform for Experimentation with Commercial Mobile Broadband Networks. MobiCom 2017, pages 70–78, 2017. http://doi.acm.org/10.1145/3117811.3117812.

[118] Utkarsh Goel, Moritz Steiner, Mike P. Wittie, Martin Flack, and Stephen Ludin. Measuring What is Not Ours: A Tale of 3^{rd} Party Performance. In Mohamed Ali Kaafar, Steve Uhlig, and Johanna Amann, editors, *Passive and Active Measurement*, pages 142–155. Springer International Publishing, 2017.

[119] Cristina Marquez, Marco Gramaglia, Marco Fiore, Albert Banchs, Cezary Ziemlicki, and Zbigniew Smoreda. Not All Apps Are Created Equal: Analysis of Spatiotemporal Heterogeneity in Nationwide Mobile Service Usage. CoNEXT 2017, 2017. http://doi.acm.org/10.1145/3143361.3143369.

[120] WhatsApp Encryption Overview. https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf. [Online; accessed 06-March-2018].

[121] iOS 11: iOS Security Guide. https://www.apple.com/business/docs/iOS_Security_Guide.pdf. [Online; accessed 06-March-2018].

[122] J. Rosenberg, R. Mahy, P. Matthews and D. Wing. RFC 5389 - Session Traversal Utilities for NAT (STUN). Technical report, Internet Engineering Task Force (IETF), 2008.

[123] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The Secure Real-time Transport Protocol (SRTP). RFC 3711 (Proposed Standard), March 2004. https://dx.doi.org/10.17487/RFC3711.

[124] Robert Birke, Marco Mellia, Michele Petracca, and Dario Rossi. Experiences of voip traffic monitoring in a commercial ISP. *International Journal of Network Management*, 20(5):339–359, 2010. https://doi.org/10.1002/nem.758.

[125] Cedric Aoun. Identifying intra-realm calls and Avoiding media tromboning. Internet-Draft draft-aoun-midcom-intrarealmcalls-00, Internet Engineering Task Force, February 2002. Work in Progress.

[126] Kolmogorov–Smirnov Test. The Concise Encyclopedia of Statistics, pages 283–287, 2008. https://doi.org/10.1007/978-0-387-32833-1_214.

[127] Ronald L. Wasserstein and Nicole A. Lazar. The ASA's Statement on p-Values: Context, Process, and Purpose. *The American Statistician*, 70(2):129–133, 2016. https://doi.org/10.1080/00031305.2016.1154108.

[128] Arturo Filastò and Jacob Appelbaum. OONI: Open Observatory of Network Interference. USENIX FOCI 2012. https://www.usenix.org/conference/foci12/workshop-program/presentation/filastò.

[129] Ooni - ooni: Open observatory of network interference. https://ooni.torproject.org/.

[130] Karsten Loesing, Steven J. Murdoch, and Roger Dingledine. A case study on measuring statistical data in the Tor anonymity network. In *Proceedings of the Workshop on Ethics in Computer Security Research (WECSR 2010)*, LNCS. Springer, January 2010.

[131] Citizenlab. citizenlab/test-lists. https://github.com/citizenlab/test-lists, March 2018.

[132] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. An in-depth study of lte: effect of network protocol and application behavior on performance. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 363–374. ACM, 2013.

[133] Safari Khatouni, A.; Mellia, M.; Ajmone Marsan, M.; Alfredsson, S.; Karlsson, J.; Brunstrom, A.; Alay, Ö; Lutu, A.; Midoglu, C.; Mancuso, V.;. Speedtest-like measurements in 3g/4g networks: the monroe experience. In *29th International Teletraffic Congress - ITC29*, pages 4–8, September 2017.

[134] Arash Molavi Kakhki, Abbas Razaghpanah, Anke Li, Hyungjoon Koo, Rajesh Golani, David Choffnes, Phillipa Gill, and Alan Mislove. Identifying traffic differentiation in mobile networks. In *Proceedings of the 2015 Internet Measurement Conference*, IMC '15, pages 239–251, New York, NY, USA, 2015. ACM.

[135] Vasilis Ververis, George Kargiotakis, Arturo Filastò, Benjamin Fabian, and Afentoulis Alexandros. Understanding internet censorship policy: The case of greece. In *5th USENIX Workshop on Free and Open Communications on the Internet (FOCI 15)*, Washington, D.C., 2015. USENIX Association.

[136] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Rfc 6182 - architectural guidelines for multipath tcp development, 2011.

[137] Nikravesh, A. and Guo, Y. and Qian, F. and Mao, Z. M. and Sen, S. An in-depth understanding of multipath tcp on mobile devices: Measurement and system design. MobiCom '16, pages 189–201, New York, NY, USA, 2016. ACM.

[138] Kevin Fall. A delay-tolerant network architecture for challenged internets. SIGCOMM '03, pages 27–34, New York, NY, USA, 2003. ACM.

[139] Safari Khatouni, A.; Ajmone Marsan, M. and Mellia, M. Delay tolerant video upload from public vehicles. SmartCity'16, pages 213–218. INFOCOM Workshop, 2016.

[140] A. S. Khatouni, M. A. Marsan, M. Mellia, and R. Rejaie. Adaptive schedulers for deadline-constrained content upload from mobile multihomed vehicles. In *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pages 1–6, June 2017.

[141] Junxian Huang, Feng Qian, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 225–238, New York, NY, USA, 2012. ACM.

[142] Huang, Junxian and Qian, Feng and Gerber, Alexandre and Mao, Z. Morley and Sen, Subhabrata and Spatscheck, Oliver. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 225–238, New York, NY, USA, 2012. ACM.

[143] Junxian Huang, Qiang Xu, Birjodh Tiwana, Z. Morley Mao, Ming Zhang, and Paramvir Bahl. Anatomizing application performance differences on smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 165–178, New York, NY, USA, 2010. ACM.

[144] Bruno Miguel Sousa, Kostas Pentikousis, and Marilia Curado. Multihoming management for future networks. *Mobile Networks and Applications*, 16(4):505–517, Aug 2011.

[145] Brett D. Higgins, Azarias Reda, Timur Alperovich, Jason Flinn, T. J. Giuli, Brian Noble, and David Watson. Intentional networking: Opportunistic exploitation of mobile network diversity. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 73–84, New York, NY, USA, 2010. ACM.

[146] Deng, S. and Netravali, R. and Sivaraman, A. and Balakrishnan, H. Wifi, lte, or both?: Measuring multi-homed wireless internet performance. IMC '14, pages 181–194, New York, NY, USA, 2014. ACM.

[147] Rahmati, A. and Zhong, L. Context-for-wireless: Context-sensitive energy-efficient wireless data transfer. MobiSys '07, pages 165–178, New York, NY, USA, 2007. ACM.

[148] Rathnayake, U.; Petander, H. and Ott, M. Emune: Architecture for mobile data transfer scheduling with network availability predictions. *Springer US*, 2012-04.

[149] Riiser, H.; Vigmostad, P.; Griwodz, C. and Halvorsen, P. Commute path bandwidth traces from 3g networks: Analysis and applications. MMSys '13, pages 114–118, New York, NY, USA, 2013. ACM.

[150] Chen, Y.; Nahum, E. M.; Gibbens, R. J.; Towsley, D. and Lim, Y. Characterizing 4g and 3g networks: Supporting mobility with multi-path tcp. Technical report, UMass Amherst Technical Report, 2012.

[151] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong. Mobile data offloading: How much can wifi deliver? *IEEE/ACM Transactions on Networking*, 21(2):536–550, April 2013.

[152] Vladimir Bychkovsky, Bret Hull, Allen Miu, Hari Balakrishnan, and Samuel Madden. A measurement study of vehicular internet access using in situ wi-fi networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, MobiCom '06, pages 50–61, New York, NY, USA, 2006. ACM.

[153] Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. How hard can it be? designing and implementing a deployable multipath TCP. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 399–412, San Jose, CA, 2012. USENIX Association.

[154] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen. Streaming high-quality mobile video with multipath tcp in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, 15(9):2345–2361, Sept 2016.

[155] Yung-Chih Chen, Yeon-sup Lim, Richard J. Gibbens, Erich M. Nahum, Ramin Khalili, and Don Towsley. A measurement-based study of multipath tcp performance over wireless networks. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 455–468, New York, NY, USA, 2013. ACM.

[156] Qiuyu Peng, Minghua Chen, Anwar Walid, and Steven Low. Energy efficient multipath tcp for mobile devices. In *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '14, pages 257–266, New York, NY, USA, 2014. ACM.

[157] Ana Nika, Yibo Zhu, Ning Ding, Abhilash Jindal, Y. Charlie Hu, Xia Zhou, Ben Y. Zhao, and Haitao Zheng. Energy and performance of smartphone

radio bundling in outdoor environments. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 809–819, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[158] Yeon-sup Lim, Yung-Chih Chen, Erich M. Nahum, Don Towsley, Richard J. Gibbens, and Emmanuel Cecchet. Design, implementation, and evaluation of energy-aware multi-path tcp. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '15, pages 30:1–30:13, New York, NY, USA, 2015. ACM.

[159] Yeon-sup Lim, Yung-Chih Chen, Erich M. Nahum, Don Towsley, and Richard J. Gibbens. How green is multipath tcp for mobile devices? In *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, AllThingsCellular '14, pages 3–8, New York, NY, USA, 2014. ACM.

[160] Bo Han, Feng Qian, Lusheng Ji, and Vijay Gopalakrishnan. Mp-dash: Adaptive video streaming over preference-aware multipath. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, pages 129–143, New York, NY, USA, 2016. ACM.

[161] Bo Han, Pan Hui, V.S. Anil Kumar, Madhav V. Marathe, Guanhong Pei, and Aravind Srinivasan. Cellular traffic offloading through opportunistic communications: A case study. In *Proceedings of the 5th ACM Workshop on Challenged Networks*, CHANTS '10, pages 31–38, New York, NY, USA, 2010. ACM.

[162] Bo Han, Pan Hui, and Aravind Srinivasan. Mobile data offloading in metropolitan area networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 14(4):28–30, November 2010.

[163] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, and V. Conan. Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–10, June 2011.

[164] O. B. Yetim and M. Martonosi. Adaptive delay-tolerant scheduling for efficient cellular and wifi usage. In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pages 1–7, June 2014.

[165] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. Augmenting mobile 3g using wifi. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 209–222, New York, NY, USA, 2010. ACM.

[166] Zaharia, M. A. and Keshav, S. Fast and optimal scheduling over multiple network interfaces. Technical report, University of Waterloo, 2007.

[167] Moo-Ryong Ra, Jeongyeup Paek, Abhishek B. Sharma, Ramesh Govindan, Martin H. Krieger, and Michael J. Neely. Energy-delay tradeoffs in smartphone applications. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 255–270, New York, NY, USA, 2010. ACM.

[168] Lutu, A.; Raj Siwakoti, Y.; Alay, Ö; Baltrūnas, Džiugas and Elmokashfi, Ahmed. The good, the bad and the implications of profiling mobile broadband coverage. *Computer Networks*, 2016.

[169] Ashkan Nikravesh, David R. Choffnes, Ethan Katz-Bassett, Z. Morley Mao, and Matt Welsh. Mobile Network Performance from User Devices: A Longitudinal, Multidimensional Analysis. In *Procs. of PAM*, 2014.

[170] Ahuja, R. K.; Magnanti, T. L. and Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[171] James Orlin. A faster strongly polynomial minimum cost flow algorithm. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 377–387, New York, NY, USA, 1988. ACM.

[172] IBM ILOG CPLEX Optimization Studio 12.6.0.0.

[173] Magharei, N. and Rejaie, R. Adaptive receiver-driven streaming from multiple senders. *Multimedia Systems*, 11(6):550–567, 2006.

[174] Sipat Triukose, Sebastien Ardon, Anirban Mahanti, and Aaditeshwar Seth. Geolocating ip addresses in cellular data networks. In Nina Taft and Fabio Ricciato, editors, *Passive and Active Measurement*, pages 158–167, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[175] Larry L. Peterson and Bruce S. Davie. *Computer Networks, Fifth Edition: A Systems Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2011.

[176] Alison L Gibbs and Francis Edward Su. On Choosing and Bounding Probability Metrics. *International statistical review*, 70(3):419–435, 2002.

[177] A. D'Alconzo, A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer. A Distribution-Based Approach to Anomaly Detection and Application to 3G Mobile Traffic. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–8, Nov 2009.